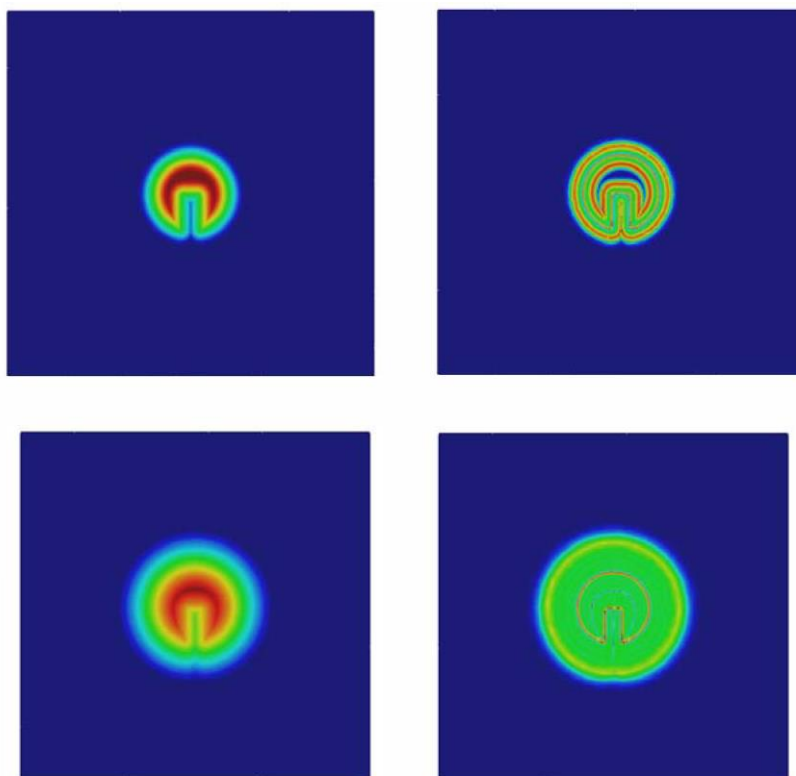


# El método de Level Set: Revisión y nuevas propuestas

A. Valls  
J. García  
E. Oñate



# **El método de Level Set: Revisión y nuevas propuestas**

A. Valls  
J. García  
E. Oñate



Register for free at <https://www.scipedia.com> to download the version without the watermark

**Publicación CIMNE N°-254, Julio 2004**

## 0. Tabla de Contenidos

Sumario.....	3
1. Introducción.....	4
2. El método Level Set.....	4
2.1 Posibles definiciones de la función Level Set.....	7
2.1.1 Definición de la función distancia en el instante inicial.....	8
2.1.2 Tiempo de cruce.....	10
2.1.3 Reinicialización de la función Level Set en cada paso de tiempo.....	11
2.2 Ejemplo teórico.....	13
3. The Narrow Band Level Set Method.....	15
3.1 El tubo.....	15
3.2 Barreras.....	16
4. Conservación de la masa.....	16
4.1 Balance de masa/volumen integral.....	16
4.2 Multiplicadores de Lagrange.....	17
4.3 Calculus ToolBox.....	18
5. Ecuaciones de Hamilton-Jacobi.....	22
5.1 Conexión con leyes conservativas.....	22
6. Implementación numérica.....	23
6.1 Esquema integración numérica ecuación de transporte.....	24
6.1.1 Formulación de la ecuación de transporte utilizando Calculo Finitesimal.....	24
6.1.2 Integración Temporal.....	26
6.1.3 Discretización Espacial.....	26
6.1.4 Ecuación de transporte discretizada por Calculo Finitesimal.....	29
6.1.5 Comentarios sobre la ecuación de reinicialización.....	29
7. Implementación algorítmica.....	30
8. Ejemplos.....	34
8.1 Rotación sólido rígido disco de Zalesak.....	34
8.2 Transporte burbuja.....	41
8.3 Elevación burbuja de gas en líquido.....	45
8.4 Sector circular.....	47
8.5 Rotación sólido rígido esfera Zalesak.....	50
8.6 Superficie Libre.....	55
9. Conclusiones.....	62
10. Referencias.....	64

Register for free at <https://www.scipedia.com> to download the version without the watermark

## Sumario

El método Level Set utilizado en un amplio espectro de problemas de distinta índole, como la simulación de interfaces entre fluidos, el tratamiento de imágenes, la combustión,...tiene como principal objetivo la simulación numérica de la evolución de interfaces entre distintas regiones en contacto.

El presente estudio es una revisión y aportación de nuevas propuestas para el método Level Set, en particular para su aplicación a la interfaz entre fluidos.

Este método formulado independiente del tipo de método utilizado para la integración numérica de las ecuaciones en derivadas parciales que gobiernan el fenómeno a simular, disfruta de gran aceptación en el mundo de las diferencias finitas y existe abundante literatura al respecto.

En este estudio se ha considerado la combinación del método Level Set y el método de Elementos Finitos no tan usual como en el caso anterior, dotando a este estudio de cierta singularidad.

El estudio se ha organizado en 9 secciones o capítulos. En capítulos del 1 al 5 se presenta una introducción teórica al método Level Set con todo el rigor matemático. Se introducen también las últimas ideas desarrolladas en este campo. A lo largo de estos capítulos se propone a modo de ejemplo la aplicación del método Level Set a la interfaz entre dos fluidos.

Los capítulos 6 y 7 se presentan los detalles computacionales de este método así como los detalles de los algoritmos necesarios para la aplicación del método de Level Set. También se hace una revisión de la discretización por el método de los Elementos Finitos estabilizados de la ecuación de transporte, necesaria para la aplicación del método.

Los capítulos 8 y 9 recogen varios resultados y conclusiones de la aplicación de este método a distintos ejemplos en dos y tres dimensiones, algunos de ellos presentes en gran parte de la literatura existente.

Register for free at <https://www.scipedia.com> to download the version without the watermark

## 1. Introducción

El problema de simular la evolución de la interfaz entre dos o mas regiones en contacto aparece en un amplio campo de problemas físicos e ingenieriles, como la localización de la superficie libre en un fluido, el transporte de contaminantes, el relleno de moldes, el crecimiento de superficies cristalinas minerales, la simulación de combustión, el tratamiento de imágenes...[6,13,17] Las características comunes en estos problemas plantean cuales son las necesidades que debe satisfacer un método numérico capaz de poder resolverlos satisfactoriamente. Entre estas necesidades cabe destacar:

- Capacidad de reproducir cambios topológicos importantes en la interfaz. Durante la evolución de la interfaz es común la partición de la interfaz en varias partes disconexas o la unión de distintos frentes.
- Formación de crestas y singularidades.
- Posibilidad de simular fenómenos 2-D y 3-D.
- Coste computacional asumible respecto al tiempo total de computación del problema.

El objetivo de este estudio es particularizar las técnicas y avances recientes en la simulación de la evolución de frentes o interfaces para el caso de flujos incompresibles con dos fases, por ejemplo aire/agua.

En particular se enfatiza el estudio de la evolución de la interfaz desde un punto de vista euleriano, mas concretamente basado en las ideas del método Level Set (LSM).

El método Level Set ofrece varias ventajas respecto otros métodos numéricos existentes para resolver este mismo problema como el método Marker Particle (MP) o el método Volume of Fluid (VOF).

Las ventajas principales del método Level Set son el hecho que es trivialmente generalizable a dimensiones superiores y de recoge de forma “natural” los cambios topológicos de la interfaz. A diferencia de otros métodos existentes, LSM captura las crestas y las colisiones entre las partículas del frente sin necesidad de costosos y complejos algoritmos que generalmente dependen de la precisión fijada por el usuario en la resolución.

El método Level Set es independiente del tipo de técnica numérica utilizada para resolver las ecuaciones en derivadas parciales (EDP's) involucradas en el problema. En este estudio la técnica numérica utilizada para la integración numérica de EDP's es el método de los Elementos Finitos (MEF) estabilizados. La combinación del MEF con LSM transfiere a este estudio un carácter singular dado que la mayor de la literatura existente sobre el uso del LSM en la simulación de la evolución de la interfaz entre dos fluidos incompresibles utiliza el método de Diferencias Finitas. Una primera ventaja a destacar de la combinación de MEF con LSM es el poder resolver este problema en dominios complejos donde la generación de una discretización espacial estructurada sea muy compleja o imposible.

## 2. El método Level Set

En esta sección se presentan los aspectos generales del método Level Set adaptado al caso de flujos incompresibles con dos fases [13,17,19]. Como se comento en la introducción el método Level Set es independiente de la técnica numérica utilizada para la integración numérica de ecuaciones diferenciales. Es por este motivo que su introducción de forma aislada es posible. En el apartado 6 se presentan las técnicas numéricas utilizadas para la solución de ecuaciones. Así se revisan también cuales son las condiciones deseables de una función Level Set, continuidad, derivabilidad, suavidad,...

Sea  $\Omega$  un dominio computacional por el que circula un flujo con dos fases, una líquida y otra gaseosa, a modo de ejemplo. Sea  $\gamma_0$  la hipersuperficie (esto es, una curva en  $\mathbb{R}^2$  y una superficie en  $\mathbb{R}^3$ ) que define el contacto entre las dos fases del flujo. Supongamos que es posible construir un campo escalar variable en el tiempo cumpliendo:

$$\phi(x, t) = \begin{cases} > 0 & x \in \text{liquido} \\ = 0 & x \in \gamma_0 \\ < 0 & x \in \text{gas} \end{cases} \quad (2.1)$$

$$\gamma_0 = \{x \mid \phi(x, 0) = 0\} \quad (2.2)$$

Un campo escalar  $\phi(x, t)$  que satisfaga las propiedades anteriores se denomina función Level Set. Se deduce de (2.2) que la función Level Set cumple que su curva de nivel cero es igual a la interfaz entre el líquido y el gas. Se llama *cero Level Set* a la curva de nivel en cero de la función Level Set.

La idea básica de método Level Set es recuperar la interfaz como la curva de nivel cero de la función Level Set. Simular la evolución temporal de la interfaz es equivalente a evolucionar temporalmente la función Level Set y localizar su curva de nivel cero.

Una primera objeción al método Level Set es el aumentar la dimensión de problema con el consecuente coste computacional asociado. Esta dificultad obliga a una implantación computacional optimizada que recibe el nombre Fast Level Set Method o Narrow Band Level Set. En la sección 3 se presenta una descripción de este método.

El LSM proporciona una descripción implícita de la interfaz. Este hecho permite el cálculo de magnitudes matemáticas asociadas a la interfaz de forma natural sin necesidad de ningún artificio adicional. Así la normal unitaria de la interfaz, orientada del gas al líquido, y la curvatura de la interfaz se pueden deducir fácilmente en términos de la función Level Set  $\phi(x, t)$ :

$$n = \frac{\nabla \phi}{|\nabla \phi|} \quad \kappa = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \bigg|_{\phi=0} \quad (2.3)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

A continuación se considera la velocidad del fluido como:

$$u = \begin{cases} u_l & \phi > 0 \\ u_g & \phi \leq 0 \end{cases} \quad (2.4)$$

Como la interfaz se desplaza con las partículas del fluido, la evolución de  $\phi(x, t)$  bien determinada por la siguiente ecuación hiperbólica (ecuación de primer orden de Hamilton-Jacobi):

$$\frac{D\phi}{Dt} = 0 \Leftrightarrow \frac{\partial \phi}{\partial t} + u \nabla \phi = 0 \quad (2.5)$$

Donde  $\frac{D}{Dt}$  denota la derivada material. En (2.5) se recoge la idea que la variación temporal la función Level Set es nula desde un punto de vista Lagrangiano, es decir, que no se permite el cambio de fase de las partículas de fluido. Toda partícula que inicialmente es líquida lo será durante toda la evolución temporal.

Como la ecuación es hiperbólica se debe imponer condición de contorno en el contorno de entrada de flujo, definido como:

$$\Gamma_{\text{inf}} = \{x \in \partial\Omega \mid u \cdot n < 0\} \quad (2.6)$$

Donde  $n$  es la normal exterior a  $\partial\Omega$ .

La condición de contorno es:

$$\phi = \bar{\phi} \quad \text{en } \Gamma_{\text{inf}} \times (t_0, t_f) \quad (2.7)$$

El valor de  $\bar{\phi}$  es positivo si hay admisión de líquido en algún punto de  $\Gamma_{\text{inf}}$ .

La condición inicial para (2.5) es

$$\phi(x, 0) = \phi_0(x) \quad (2.8)$$

Con  $\phi_0(x)$  definida según la posición inicial de interfaz, como (2.1).

En resumen la función Level Set es la solución del siguiente problema:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + u \nabla \cdot \phi &= 0 \quad \text{en } \Omega \times (t_0, t_f) \\ \phi &= \bar{\phi} \quad \text{en } \Gamma_{\text{inf}} \times (t_0, t_f) \\ \phi(x, 0) &= \phi_0(x) \quad \text{en } \Omega \end{aligned} \quad (2.9)$$

Otra presentación más intuitiva sobre la ecuación que gobierna la evolución de  $\phi(x, t)$  es considerar el método de las características. Sin pérdida de generalidad suponemos que el dominio  $\Omega$  es bidimensional [19].

Sea  $x(s, t), y(s, t)$  una parametrización de la interfaz  $\gamma$ , en tiempo  $t$ .

$$\gamma(s, t) = (x(s, t), y(s, t)) \quad (2.10)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

La evolución temporal de  $(x, y)$  viene descrita por las siguientes dos ecuaciones

$$\begin{aligned} \frac{dx(s, t)}{dt} &= u(x(s, t), y(s, t)) \\ \frac{dy(s, t)}{dt} &= v(x(s, t), y(s, t)) \end{aligned} \quad (2.11)$$

Como  $\phi(x(s, t), y(s, t))$  está definido para ser cero en la interfaz (la curva de nivel cero de  $\phi$  es la superficie que define la interfaz), se tiene que

$$\frac{d\phi(x(s, t), y(s, t), t)}{dt} = \frac{d\phi}{dx} \frac{dx}{dt} + \frac{d\phi}{dy} \frac{dy}{dt} + \frac{d\phi}{dt} = \phi_t + u\phi_x + v\phi_y = 0 \quad (2.12)$$

Veamos ahora como son las soluciones analíticas de (2.9). Para simplificar el estudio, se restringe al caso 1-D. Esto es  $\phi = \phi(x, t) \quad x \in \mathbb{R}, \quad t \in [0, T]$  y sin pérdida de generalidad se puede asumir que el valor prescrito en el contorno de entrada es nulo.

En esta situación:

Se cumple que la velocidad  $u$  no depende de  $\phi$ , y asumimos que es constante en  $\Delta t$ .

En  $\Delta t$  fijo.

$$\begin{aligned}\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} &= 0 \quad u = \text{cte} \\ \phi(x, 0) &= \phi_0(x)\end{aligned}\tag{2.13}$$

Se cumple que  $\phi = \phi_0(x - ut)$  es solución de (2.13).

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= \frac{\partial}{\partial t}(\phi_0(x - ut)) = -u \frac{\partial \phi_0}{\partial x} \\ \frac{\partial \phi}{\partial x} &= \frac{\partial}{\partial x}(\phi_0(x - ut)) = \frac{\partial \phi_0}{\partial x} \\ \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} &= -u \frac{\partial \phi_0}{\partial x} + u \frac{\partial \phi_0}{\partial x} = 0\end{aligned}\tag{2.14}$$

Se deduce de  $\phi = \phi_0(x - ut)$  que el comportamiento de la solución es simplemente el transporte o traslación de la configuración inicial  $\phi_0$  una distancia  $u\Delta t$ . Véase figura 1.

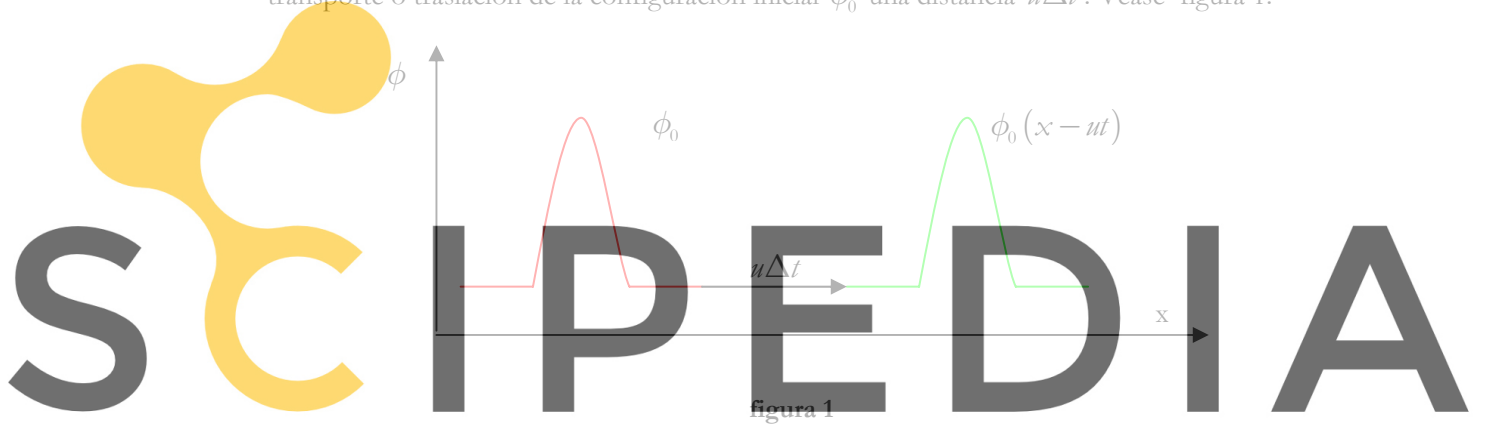


figura 1

Estas mismas ideas se pueden presentar en varias dimensiones obteniendo el mismo tipo de soluciones.

Register for free at <https://www.scipedia.com> to download the version without the watermark

En conclusión, en el caso de un campo de velocidades constante, la solución de la ecuación (2.5) es el transporte la configuración inicial. La ecuación (2.5) es denominada también como ecuación de transporte. En el caso de un velocidad no constante, la solución de (2.5) será sucesión de soluciones obtenidas en cada incremento de tiempo a velocidad constante.

En el caso de trabajar con fluidos incompresibles se cumple que la ecuación de conservación de masa se reduce a

$$\nabla \cdot u = 0\tag{2.15}$$

Así la ecuación de transporte se puede describir en forma de divergencia como:

$$\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = \frac{\partial \phi}{\partial t} + \nabla \cdot (u\phi)\tag{2.16}$$

## 2.1 Posibles definiciones de la función Level Set

Uno de los pasos fundamentales en el método Level Set es la definición de la función Level Set. A continuación se presentan las propiedades deseables para una función Level Set.

A priori una función escalón (véase **figura 2**) es aceptable como función Level Set



$$\phi(x, t) = \begin{cases} +1 & x \in \text{liquido} \\ 0 & x \in \gamma_0 \\ -1 & x \in \text{gas} \end{cases} \quad (2.17)$$

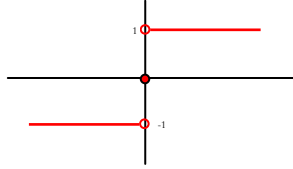


figura 2

La elección de una función escalón tiene asociada grandes problemas numéricos. Es sabido que pequeñas oscilaciones en el entorno de grandes gradientes se transportan durante la resolución de la ecuación (2.9). Estas oscilaciones se pueden propagar y amplificar distorsionando la forma del frente.

Es preferible usar una función suave, es decir, diferenciable y con gradientes suaves. Varias funciones han sido propuestas, pero salvo pequeñas variaciones, la función distancia con signo a la interfaz presenta mejores propiedades.

Sea  $\phi$  una función cumpliendo (2.1) y (2.2) que satisface:

$$\|\nabla \phi\| = 1 \quad x \in \Omega \quad (2.18)$$

Entonces  $\phi$  es una función distancia con signo a  $\gamma$ . Para justificar esta propiedad considerar el siguiente argumento:

Por (2.2) se cumple que la interfaz es la curva cero Level Set de  $\phi$ , esto implica que:

$$\nabla \phi \mid n \quad n = \text{normal a } \gamma. \text{ Sea } x \in \Omega$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

$$\phi(x) = \pm \text{dist}(x, \gamma) = \pm \min \{ \|x - \tilde{y}\| \mid \tilde{y} \in \gamma \}$$

$$\text{Como } \gamma \text{ es compacto en } \mathbb{R}^d \Rightarrow \exists \tilde{y} \in \gamma \mid \text{dist}(x, \gamma) = \|x - \tilde{y}\|$$

Se cumple, por ser  $\mathbb{R}^d$  estrictamente normado:

$$x - \tilde{y} \mid n \Rightarrow x - \tilde{y} \mid \nabla \phi|_{\tilde{y}} \Rightarrow x - \tilde{y} = \lambda \nabla \phi|_{\tilde{y}} \quad \lambda \in \mathbb{R}$$

$$\|x - \tilde{y}\| = |\lambda| \|\nabla \phi|_{\tilde{y}}\| = |\phi(x)| \|\nabla \phi|_{\tilde{y}}\| \underset{\|\nabla \phi\|=1}{\Rightarrow} \|x - \tilde{y}\| = |\phi(x)| = \text{dist}(x, \gamma)$$

$$\phi(x) = \pm \text{dist}(x, \gamma)$$

qed

Tenemos entonces que el problema a resolver es la definición de una función distancia con signo en el dominio  $\Omega$  que cumpla (2.1) y (2.2).

Existen dos etapas en la definición de la función distancia con signo:

### 2.1.1 Definición de la función distancia en el instante inicial.

La localización de la interfaz vendrá determinada por la diferencia en las propiedades de las fases existentes en el flujo como la densidad, viscosidad, ... Sea  $\eta$  esta propiedad,

$$\eta = \begin{cases} \eta_{\text{liquido}}(x, t) & x \in \Omega_t \\ \eta_{\text{gas}}(x, t) & x \in \Omega \setminus \Omega_t \end{cases} \quad (2.19)$$

$$\Omega_t = \{x \in \Omega \mid \phi(x, t) > 0\}$$

La definición de la función Level Set como función distancia a partir de estos datos iniciales se puede realizar de varias formas, a continuación se detallan dos posibles alternativas:

a) *Algoritmo geométrico puro:*

Para el cálculo de la distancia de un nodo,  $i$ , al frente se considera una discretización de la interfaz en triángulos planos,  $p$ , (en el caso 2-D, segmentos) y se calcula la distancia perpendicular del nodo a todos los triángulos de interfaz, quedándonos la mínima.

Este algoritmo presenta dificultades en definir una distancia topológicamente coherente con la geometría del modelo. Se han obtenidos problemas con geometrías en formas de  $\mathbb{U}$ . Adicionalmente este algoritmo necesitará de un proceso de suavizado para mejorar las propiedades numéricas de la función Level Set obtenida. En realidad con este algoritmo después del suavizado no se obtiene una función distancia con signo sino una función de ella.  $\phi(x, t) = \text{sign}(\phi) f(d(x, t))$ . Donde  $d(x, t)$  denota la distancia al frente.

b) *Algoritmo matemático puro:*

Este algoritmo plantea una resolución del problema absolutamente diferente. Dada una función Level Set que no defina una función distancia con signo, se reinicializa de la siguiente manera [13,17,19,20]  
Considerar la EDP

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi)(1 - \|\nabla d\|) \quad (2.20)$$

Con condición inicial

Register for free at <https://www.scipedia.com> to download the version without the watermark

$$\text{Donde } \text{sign}(\phi) = \begin{cases} -1 & \phi < 0 \\ 0 & \phi = 0 \text{ y } \tau \text{ es un tiempo artificial.} \\ +1 & \phi > 0 \end{cases}$$

La soluciones estacionaras de (2.20) son funciones distancia. Vemos este hecho: Queremos que se cumpla (2.18) equivalentemente:

$$1 - \|\nabla d\| = 0 \quad (2.21)$$

Supongamos que (2.21) no se cumple, entonces:

$$1 - \|\nabla d\| = r \quad r \neq 0 \quad (2.22)$$

Imponemos que

$$\frac{\partial d}{\partial \tau} = r \quad (2.23)$$

La solución estacionaria de (2.23) cumple

$$\frac{\partial d_{\text{steady}}}{\partial \tau} = 0 \Rightarrow r = 0 \Rightarrow \|\nabla d\| = 1 \quad (2.24)$$

Además como  $sign(\phi) = 0$  cuando  $\phi = 0$  se tiene que el cero Level Set de  $d(x, \tau)$  es el mismo de  $\phi(x)$ . Por lo tanto simplemente se debe calcular la solución estacionaria de (2.20) y sustituir  $\phi(x)$  por  $d(x, \tau_{steady})$ . Si se considera que en realidad no se necesita la función Level Set definida como función distancia en todo el dominio (véase Narrow Band Level Set method) sino en una zona cercana a la interfaz, pongamos de ancho  $2\varepsilon$ , se llega al estado estacionario cerca de interfaz cuando se cumple:

$$\|\nabla d\| = 1 \quad \text{para} \quad |d| < \varepsilon \quad (2.25)$$

Además se cumple que este algoritmo empieza a reinicializar la función Level Set cerca de frente en primer lugar. Veamos este hecho: Rescribimos la ecuación (2.20) de la siguiente forma:

$$\begin{aligned} \frac{\partial d}{\partial \tau} + \omega \cdot \nabla d &= sign(\phi) \\ \omega &= sign(\phi) \frac{\nabla d}{\|\nabla d\|} \end{aligned} \quad (2.26)$$

Es claro que (2.26) es una ecuación hiperbólica no lineal con velocidad característica  $\omega$  apuntando hacia el exterior de la interfaz (del gas hacia el liquido), según la normal al frente. Por consiguiente  $d$  será reinicializada a  $\|\nabla d\| = 1$  cerca de la interfaz en primer lugar.

Al ser solo necesario definir la función Level Set como función distancia cerca de la interfaz se puede usar un numero fijo de pasos para asegura la propiedad de función distancia cerca de la interfaz. Por ejemplo si el paso de pseudo-tiempo es  $\Delta\tau$  y el ancho de banda cercano a interfaz para el que se quiere definir la función Level Set como función distancia es  $2\varepsilon$ , se puede parar el proceso iterativo después de  $\varepsilon/\Delta\tau$  iteraciones [11,12,15].

Es importante resaltar que (2.26) es una EDP de tipo similar a la ecuación de transporte, esto permite utilizar el mismo esquema de integración para ambas ecuaciones.

### 2.1.2 Tiempo de cruce

Consideremos la ecuación (2.26) nuevamente

$$\begin{aligned} \frac{\partial d}{\partial \tau} + \omega \cdot \nabla d &= \frac{\partial d}{\partial \tau} + sign(\phi) \frac{\nabla d}{\|\nabla d\|} \cdot \nabla d = \\ &= \frac{\partial d}{\partial \tau} + sign(\phi) \vec{N} \cdot \nabla d = sign(\phi) \end{aligned} \quad (2.27)$$

Donde  $\vec{N} = \frac{\nabla d}{\|\nabla d\|}$  es el vector normal a la interfaz.

Así se puede concluir que solucionar la ecuación (2.26) es equivalente a desplazar el frente según su normal unitaria, con signo, (velocidad de desplazamiento unitaria). Dado que la velocidad de desplazamiento es unitaria y ortogonal a la interfaz, el tiempo necesario para que un punto sea alcanzado por la interfaz es equivalente a la

distancia de este punto respecto la posición inicial de la interfaz. Este tiempo se llama tiempo de cruce y es equivalente a la distancia a la interfaz. Este hecho clarifica el porque los valores de la función Level Set son reinicializados en las zonas cercanas al frente en primer lugar.[13]

### 2.1.3 Reinicialización de la función Level Set en cada paso de tiempo.

Aún cuando la función Level Set inicial  $\phi(x, 0)$  sea un función distancia con signo es sabido que esta propiedad se puede perder durante la evolución temporal de (2.9). Se plantea entonces un nuevo problema, dada un a función Level Set reinicializarla para obtener una función distancia con el mismo cero Level Set.

Como se vio en el apartado anterior el *algoritmo matemático puro* soluciona este problema sin necesidad de más algoritmia.

Aún así este algoritmo puede presentar problemas en mallas muy distorsionadas con singularidades y/o esquinas. Estas casuísticas dan lugar a oscilaciones espurias que pueden deteriorar la evolución de la interfaz e introducir variaciones espurias en la masa del flujo.

A continuación se presenta una algoritmo de suavizado o reinicialización basado en ideas topológicas y geométricas, que no presenta las dificultades del método anterior. Antes de la descripción del método es necesario introducir el concepto de niveles nodales[5].

-Niveles nodales:

Dada una distribución de una función Level Set, y por lo tanto la posición del frente para todo nodo  $i$  de la malla, se asigna nivel  $l_i$  a nodo  $i$  según las siguientes reglas:

- Un nodo  $i$  pertenece a un nivel  $l_i = 0$  si esta en el liquido ( $\phi > 0$ ) y esta conectado como mínimo a un nodo que esta en el gas ( $\phi < 0$ ).
- Un nodo  $i$  pertenece a un nivel  $l_i > 0$ , si esta en el gas ( $\phi < 0$ ) y esta conectado como mínimo a un nodo de nivel  $l_i - 1$ .
- Un nodo  $i$  pertenece a un nivel  $l_i < 0$ , si esta en el liquido ( $\phi > 0$ ) y esta conectado como mínimo a un nodo de nivel  $l_i + 1$ .

Observar que los niveles así definidos son enteros.

Un vez los niveles han sido asignados, las distancias al frente  $d_i$  de los nodos que pertenecen a los niveles  $0 \leq l_i \leq 1$  son calculadas usando el *algoritmo geométrico puro* descrito anteriormente. Este cálculo se puede hacer rápidamente si solo se calculan distancias para los puntos en el entorno de cada triángulo plano que discretiza el frente. Es claro que los nodos pertenecientes a los niveles 0 y 1 pertenecen a elementos cortados por la interfaz. Definimos  $\phi(x, t) = f(d(x, t))$  para un nodo  $i$  como sigue:

$$f_i = \begin{cases} d_i & 0 \leq l_i \leq 1 \\ d_m + (l_i - 1)\bar{d} & l_i > 1 \\ d_m - l_i\bar{d} & 0 < l_i \end{cases} \quad (2.28)$$

Donde

$$\begin{aligned} \bar{d} &= \frac{1}{N} \sum_i d_i & d_m &= \max_i \{d_i\} & \forall i | 0 \leq l_i \leq 1 \\ N &= \#\{i | 0 \leq l_i \leq 1\} \end{aligned} \quad (2.29)$$

Básicamente el campo anterior es proporcional a los niveles nodales, y por consiguiente sigue la topología de la malla. Sin embargo este campo puede presentar discontinuidades o grandes gradientes y necesita de un algoritmo de suavizado.

- *Algoritmo de suavizado.*

Este algoritmo de suavizado se basa en ideas del campo del tratamiento de imágenes y del suavizado de superficies [21].

Para cada nodo  $i$  perteneciente a niveles nodales  $l_i < 0 \quad l_i > 1$

$$\begin{aligned} f_i^{n-1/2} &:= f_i^{n-1} + \lambda \partial f_i^{n-1} \\ f_i^n &:= f_i^{n-1/2} + \mu \partial f_i^{n-1/2} \quad n = 1, \dots, n_{pass} \end{aligned} \quad (2.30)$$

Donde  $n_{pass}$  es el número de pasadas de suavizado,  $\lambda$  y  $\mu$  son constantes con valores óptimos 0.6313983 y -0.6739515, respectivamente, y:

$$\partial f_i = \sum_{j \in N_i} w_{ij} (f_i - f_j) \quad w_{ij} = \left( d_{ij} \sum_{j \in N_i} \frac{1}{d_{ij}} \right)^{-1} \quad (2.31)$$

Donde  $N_i$  es el entorno del nodo  $i$ , esto es, el conjunto de nodos  $j$  conectados a  $i$ , y  $d_{ij}$  es la distancia entre los nodos  $i$  y  $j$ .

Finalmente la función Level Set suavizada se define como

$$\phi(x, t) = f^{n_{pass}}(x, t) \quad (2.32)$$

El número de pasadas de suavizado se escoge de forma heurística, el valor optimo recomendado es  $n_{pass} = \sqrt{n_p}/2$  con  $n_p$  el número total de nodos de la malla.

Resaltar que este algoritmo de suavizado se aplica únicamente a nodos en niveles nodales  $l_i < 0 \quad l_i > 1$ , esto garantiza que no se modifica la posición del frente.

Este algoritmo se ha introducido aplicado a toda la malla, es posible optimizarlo aplicándolo únicamente a una banda que contenga la interfaz (véase Narrow Band Level Set method). En la figura 3 se puede observar como se definen los niveles nodales y el resultado del algoritmo de suavizado.

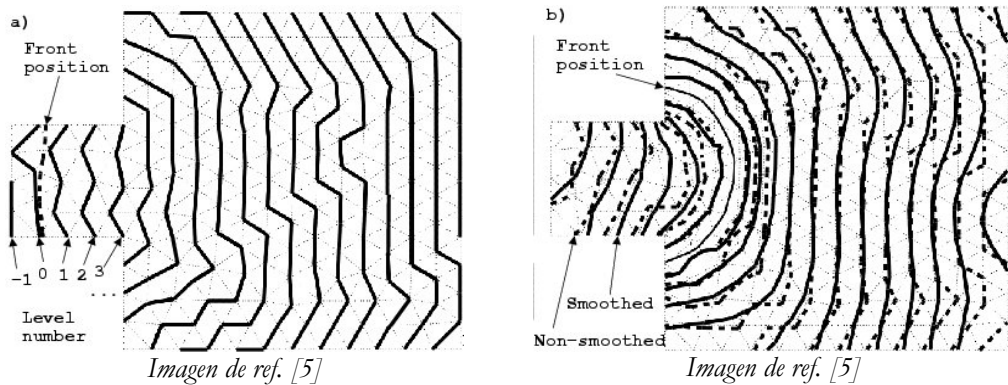


figura 3

La ventaja clara de este algoritmo frente al algoritmo matemático puro es su coste computacional. La implementación de este algoritmo no necesita de la integración de una EDP hasta su estado estacionario en cada paso de tiempo.

La gran desventaja de este algoritmo es que no realiza el suavizado donde es más necesario, a saber, en el entorno de la interfaz. Los niveles nodales 0 y 1 no son

tratados en ninguna etapa del algoritmo. Así aunque la distribución de la función Level Set sea parecida a una función distancia y suave en todo el dominio  $\Omega$ , la zona de mayor importancia y origen de errores numéricos no es tratada en ningún momento. Para evitar este problema se ha realizado una modificación del algoritmo anterior para los niveles nodales 0 y 1 como sigue:

Para los nodos  $i$  de nivel  $l_i = 0$

$$\phi_i^{n+1} = \frac{(1 + \|\nabla \phi_i^n\|) \phi_i^n}{2 \|\nabla \phi_i^n\|} \quad n = 1, \dots, n_{pass} \quad i \in \{i \in \mathbb{N} \mid l_i = 0\} \quad (2.33)$$

Para los nodos  $i$  de nivel  $l_i = 1$

$$\begin{aligned} \phi_i^{n+1} &= \frac{(1 + \|\tilde{\nabla} \phi_i^n\|) \phi_i^n}{2 \|\tilde{\nabla} \phi_i^n\|} \quad n = 1, \dots, n_{pass} \\ \tilde{\nabla} \phi_i^n &= \nabla \phi_i^n / n_i^0 \quad i \in \{i \in \mathbb{N} \mid l_i = 1\} \\ n_i^0 &= \# \{j \mid \text{nodo } j \text{ esta conectado al nodo } i \text{ y } l_j = 0\} \end{aligned} \quad (2.34)$$

El objetivo de esta modificación es suavizar la función Level Set en los niveles nodales 0 y 1 cumpliendo con dos restricciones:

- Desplazar la interfaz de forma que se conserve el volumen de líquido.
- Conseguir que la función Level Set también defina una función distancia suave para los niveles 0 y 1.

Evidentemente al aplicar esta modificación la interfaz se desplaza respecto su posición original, hecho que a priori no es en absoluto deseado, pero en cambio se obtienen un distribución de la función Level Set suave en un entorno de la interfaz.

La idea que sigue esta modificación es la ponderación entre el desplazamiento de la interfaz frente a una distribución suave de  $\phi$  y en consecuencia con mejores propiedades numéricas.

## 2.2 Ejemplo teórico

Para ejemplificar el concepto del método Level Set, se propone el siguiente caso [24]:

El dominio de estudio será un cuadrado en  $\mathbb{R}^2$ . En el centro del dominio se tiene una burbuja de aire de forma circular con velocidad de expansión  $F$  positiva, normal a la curva interfaz. Véase figura 4. En el exterior de la burbuja tenemos líquido.

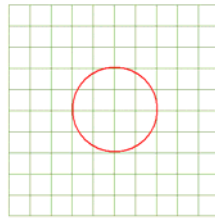


figura 4

Como el dominio tiene dimensión 2, el frente es una hipersuperficie de  $\mathbb{R}^2$ , esto es una curva.

La función Level Set  $\phi(x, t)$  será una superficie que toma valores negativos en el interior del círculo, nulos en la frontera y positivos en el exterior del círculo.

Si consideramos  $\phi(x, t)$  como la función distancia con signo a la interfaz, es decir:

$$\phi(x, t) = \pm d(x, t) \quad (2.35)$$

Donde  $d(x)$  es la distancia euclídea de un punto a la interfaz.

En consecuencia el aspecto topológico de la función  $\phi(x, t)$  será de superficie cónica con vértice en el punto  $(0, 0, r)$  donde  $r = \text{radio del círculo en el tiempo } t$ .

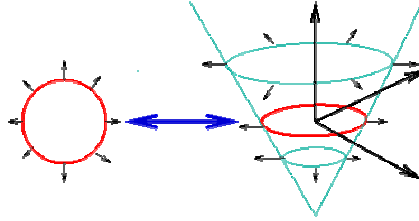


figura 5

En la figura 5, el círculo de la izquierda es la posición del frente, contenido en el plano XY. La ilustración de la derecha es la representación de la función Level Set. El frente se recupera como la intersección de la función Level Set con el plano XY  $\{(x, y, z) \in \mathbb{R}^3 \mid z = 0\}$ .

En la figura 6 se presenta una secuencia de imágenes que recogen la evolución temporal de la función Level Set y consecuentemente la expansión de la burbuja de aire.

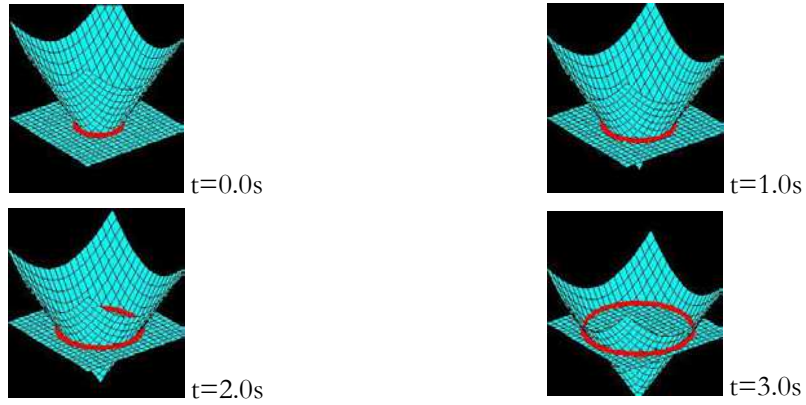


figura 6

Una de las ventajas ya comentadas del método Level Set es la captura de forma “natural” de los cambios topológicos de la interfaz sin necesidad de rutinas adicionales. Veamos por ejemplo el comportamiento del método en el caso de dos burbujas de aire en condiciones de expansión como las anteriores.





figura 7

En este caso la superficie asociada a la función Level Set es la intersección de dos conos en su parte superior (véase, figura 7).

### 3. The Narrow Band Level Set Method.

En la sección anterior se introdujo con generalidad el método del Level Set y se motivó la necesidad de una implementación computacional mejorada para reducir el tiempo efectivo de cálculo. Dado que el método Level Set es necesario aplicar con bastante frecuencia durante la evolución temporal, es vital para la resolución de problemas en tiempos razonables una modificación del método antes descrito.

A continuación se introduce el llamado Fast Level Set Method o equivalentemente el Narrow Band Level Set Method [17]. Este método es independiente del tipo de algoritmo usado para la definición de la función distancia.

#### 3.1 El tubo

Para usar el Narrow Band Level Set Method, el primer paso es la construcción de un tubo en el entorno de interfaz, donde se definirá de forma precisa (debidamente suavizada) la función Level Set.

Se construye un tubo conteniendo todos los puntos con distancia al frente menor a  $\maxDist$ , valor prefijado.

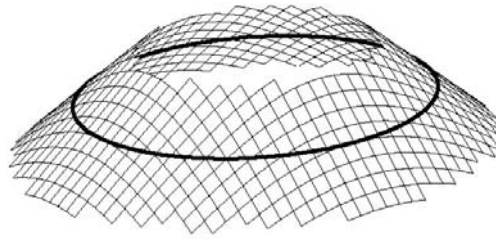


figura 8

En lugar de calcular la función Level Set en todo el dominio, se calcula únicamente en los nodos contenidos en el tubo y el resto son inicializados con valores  $\pm \maxDist$ .

El ahorro operacional del método depende directamente del ancho de banda del tubo, así si originalmente se necesitaban  $O(n^3)$  operaciones para el cálculo de la función distancia sobre toda la malla, ahora se necesitan solo  $O(nk^2)$ , con  $k$  el número de nodos extendidos mas allá de la interfaz.

Para que este método funcione es necesario que durante la evolución temporal la interfaz esté contenida en su totalidad en el interior del tubo; esto se puede garantizar actualizando la posición del tubo en cada paso de tiempo. Esta opción aunque válida, se puede mejorar computacionalmente hablando como sigue: Dado un tubo, la estrategia a seguir es realizar el máximo número de pasos de tiempo usando el mismo tubo y comprobando en cada paso de tiempo si la interfaz aún se encuentra en interior del tubo.



### 3.2 Barreras

El paso siguiente en el diseño del Narrow Band Level Set Method es desarrollar un algoritmo de detección de la proximidad de interfaz al contorno del tubo. Tomando un valor prefijado  $bowClose$ , la idea es detectar cuando la interfaz esta mas cercana a la frontera del tubo que  $bowClose$ .

La opción obvia de calcular la distancia exacta entre la interfaz y el contorno del tubo, es exageradamente costosa y consumiría la mayor parte del tiempo computacional en cada paso de tiempo. Notar que en realidad no se necesita la distancia exacta entre el frente y el contorno del tubo, únicamente comprobar si la distancia es menor a un cierto valor prefijado o no.

Utilizamos el valor de la función Level Set, si un nodo tiene un valor de  $\phi$  negativo pertenece a una de las componentes conexas en las que la interfaz divide el dominio, en caso contrario a la otra.

Si la interfaz esta a una distancia inferior a  $bowClose$  del contorno de tubo entonces existe un nodo a distancia menor que  $bowClose$  en el que ha cambiado el signo de la función Level Set. Por lo tanto, cuando inicializamos la función Level Set en el tubo tenemos las curvas de nivel de  $\phi$  entre las alturas  $\pm (maxDist-bowClose)$ . Redondeando los valores extremos de las curvas de nivel a los nodos de la malla más próximos se construye una barrera interior al tubo a una distancia  $maxDist-bowClose$  de la interfaz (o a una distancia  $bowClose$  de contorno del tubo).

Se almacenan dos conjuntos de nodos, la barrera con valores negativos de  $\phi$  y otra barrera con valores positivos de  $\phi$ . En cada paso de tiempo comprobamos si alguno de estos nodos ha cambiado su signo, en caso afirmativo reinicializamos la posición del tubo usando el ultimo cero Level-Set usado.

El número de nodos extendidos ha considerar se puede tomar como:

$$\left( \nabla \phi|_y \cdot u(y) \right) \Delta t < \phi_i \quad y \in \gamma \quad (3.1)$$

## 4. Conservación de la masa.

Uno de los problemas detectados en el uso del método Level Set es la pérdida neta (pérdida o ganancia) de masa/volumen de la función Level Set durante la evolución temporal. Este fenomeno no es únicamente imputable al método Level Set también se debe también al esquema de integración de la ecuación de Hamilton-Jacobi de primer orden y de la ecuaciones de Navier-Stokes para flujos incomprensibles. Por ejemplo es posible que las velocidades obtenidas de Navier-Stokes no sean libres de divergencia y consecuentemente el balance de masa/volumen será imposible.

A continuación se presentan diversas estrategias para solucionar, en lo posible, este problema.

En este estudio se ha optado por la propuesta 4.1 dada su sencillez y eficacia frente la propuesta en 4.2 mas costos de implementar y con resultados equivalentes.

### 4.1 Balance de masa/volumen integral

Sabemos que se debe cumplir el siguiente balance de masa/volumen en cualquier paso de tiempo [5]:

Sea  $V_{\Omega}^n$  el volumen de la región  $\Omega^- = \{x \in \Omega \mid \phi(x, t^n) \leq 0\}$ , en el tiempo  $t^n$ .

Sea  $V_{\Omega}^{n+1}$  el volumen de la región  $\Omega^- = \{x \in \Omega \mid \phi(x, t^{n+1}) \leq 0\}$ , en el tiempo  $t^{n+1}$ .

Sea  $V_{\Omega}^{inc}$  el volumen asociado al incremento de  $\phi$  en el incremento de tiempo  $t^{n+1} - t^n$ .

Entonces, se debe cumplir por incompresibilidad del flujo

$$V_{\Omega}^{n+1} = V_{\Omega}^n + V_{\Omega}^{inc} \quad (4.1)$$

Para corregir el balance de masa/volumen (4.1) se añade o se subtrae la cantidad de masa perdida o ganada de forma uniforme sobre toda la interfaz. Con este objetivo la interfaz se desplaza una distancia  $s$  según su normal para obtener el balance de masa.

Sea  $L$  la superficie que define la interfaz (curva en 2-D, línea en 1-D), se define la distancia  $s$  como

$$s = \frac{V_{\Omega}^{n+1} - (V_{\Omega}^{inc} + V_{\Omega}^n)}{L} \quad (4.2)$$

Así después de mover la interfaz la distancia  $s$  definida en (4.2), el nuevo volumen  $V_{\Omega}^{n+1}$  cumple el balance de masa/volumen

$$V_{\Omega}^{n+1} = V_{\Omega}^{inc} + V_{\Omega}^n + L \cdot s \quad (4.3)$$

En la sección 4.3 se muestran los detalles de cómo calcular volúmenes utilizando la teoría del método de Level Set.

## 4.2 Multiplicadores de Lagrange

Tenemos que la interfaz viene inicialmente definida según (2.19). Para superar los problemas numéricos asociados al transporte de una función discontinua, se ha forzado a definir la función Level Set como una función distancia con signo, en particular continua [18,19,20].

La interfaz es inmaterial y por tanto carece de espesor. Tenemos entonces una transición brusca (discontinua) de las propiedades del flujo en la interfaz. A continuación realizamos una ligera modificación en (2.19), para suavizar la transición de las propiedades de flujo. Esta modificación convierte la interfaz en una entidad material asignando un espesor.

$$\eta(\phi) = \eta_{gas} + (\eta_{gas} - \eta_{liquido}) H_{\varepsilon}(\phi)$$

$$H_{\varepsilon}(\phi) = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2} \left[ 1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\varepsilon}\right) \right] & |\phi| < \varepsilon \\ 1 & \phi > \varepsilon \end{cases} \quad (4.4)$$

Según (4.4) el espesor la interfaz es aproximadamente

$$\frac{2\varepsilon}{\|\nabla\phi\|} \quad (4.5)$$

Si tenemos  $\phi$  definida como una distancia tenemos trivialmente que el espesor de la interfaz es  $2\varepsilon$ , en particular constante durante toda la evolución temporal.

Una posible elección del espesor es  $\varepsilon = \alpha L$  donde  $L$  es un tamaño característico de los elementos de la malla. De este modo a medida que se refina la malla se reduce el espesor de interfaz.

La condición de conservación de masa será aproximadamente equivalentemente a que el espesor de la interfaz sea constante. Esta hipótesis permite poner el problema de la conservación de la masa como una restricción matemática al problema del transporte del frente.

Así, suponiendo que  $\phi$  es una función distancia cumpliendo (2.25)

Imponemos la siguiente restricción

$$\partial_{\tau} \int_{\Omega} H(\phi) = 0 \quad (4.6)$$

Y plantemos la siguiente modificación de la ecuación (2.20)

$$\begin{aligned}\frac{\partial d}{\partial \tau} &= L(\phi_0, \phi) + \lambda g(\phi) = \text{sign}(\phi)(1 - \|\nabla d\|) + \lambda g(\phi) \\ \phi(x, 0) &= \phi_0(x)\end{aligned}\tag{4.7}$$

Donde  $\lambda$  es solo función del pseudo-tiempo, determinada por,

$$\begin{aligned}\partial_\tau \int_{\Omega} H(\phi) &= \int_{\Omega} H'(\phi) \phi_\tau = \int_{\Omega} H'(\phi) (L(\phi_0, \phi) + \lambda g(\phi)) = 0 \\ &\quad - \int_{\Omega} H'(\phi) L(\phi_0, \phi) \\ \lambda &= \frac{\int_{\Omega} H'(\phi) L(\phi_0, \phi)}{\int_{\Omega} H'(\phi) g(\phi)}\end{aligned}\tag{4.8}$$

Tomando la función  $g$  como

$$g(\phi) = H'(\phi) \|\nabla \phi\| \tag{4.9}$$

Se garantiza que solo se corrige la interfaz sin modificar la propiedad de función distancia fuera de interfaz.

Notar que si (2.20) es resuelto perfectamente entonces  $\lambda$  será idénticamente nulo. Esto es debido a que  $L(\phi_0, \phi)$  será cero en las regiones donde  $H'(\phi)$  no es nulo. Discretamente esto no es cierto debido a que el cero Level Set de  $\phi_0$  puede diferir del cero Level Set de  $\phi$  debido a errores numéricos

Desde un punto de vista numérico puede resultar más cómodo y eficiente imponer la restricción (4.6) a nivel elemental. Así las ecuaciones (4.7) y (4.8) resultan:

$$\begin{aligned}\frac{\partial d}{\partial \tau} &= L(\phi_0, \phi) + \lambda_e g(\phi) \quad x \in e = \text{elemento} \\ &\quad - \int_{\Omega_e} H'(\phi) L(\phi_0, \phi) \\ \lambda_e &= \frac{\int_{\Omega_e} H'(\phi) L(\phi_0, \phi)}{\int_{\Omega_e} H'(\phi) g(\phi)}\end{aligned}\tag{4.10}$$

Esta modificación conservativa obliga a una modificación de (2.20) y únicamente se aplica en su integración numérica. Consecuentemente su implementación numérica será más costosa que la modificación de conservación-balance y no garantiza que se corrijan las pérdidas netas de masa debidas a errores numéricos en la integración de la ecuación (2.9), que gobierna la evolución del frente.

### 4.3 Calculus ToolBox

En las anteriores modificaciones conservativas es necesario calcular integrales de distintas funciones sobre varios dominios. Veamos como el método Level Set permite poder describir estas integrales de forma muy apropiada para su posterior cálculo numérico [13].

Se define, para un instante de tiempo fijo  $t_0$ ,

$$\begin{aligned}\Omega^- &= \{x \in \Omega \mid \phi(x, t_0) < 0\} \\ \Omega^+ &= \{x \in \Omega \mid \phi(x, t_0) > 0\}\end{aligned}\tag{4.11}$$

En el resto de esta sección se omite a efectos de notación la dependencia temporal de la función Level Set, dado que el instante de tiempo se considera fijo.

La función característica  $\chi^-$  de la región interior  $\Omega^-$  se define como

$$\chi^-(x) = \begin{cases} 1 & \phi(x) \leq 0 \\ 0 & \phi(x) > 0 \end{cases} \quad (4.12)$$

Donde se incluye de forma arbitraria el contorno a la región interior. La función característica  $\chi^+$  de la región exterior  $\Omega^+$  se define de forma análoga como

$$\chi^+(x) = \begin{cases} 0 & \phi(x) \leq 0 \\ 1 & \phi(x) > 0 \end{cases} \quad (4.13)$$

Incluyendo el contorno en la región interior otra vez. El contorno o interfaz definido como

$$\gamma_0 = \{x \in \Omega \mid \phi(x) = 0\} \quad (4.14)$$

El contorno es un conjunto de medida cero dado que  $\phi$  es continua y  $\Omega$  esta acotado y en consecuencia se cumple

$$\begin{aligned} \int_{\Omega} f(x) dx &= \int_{\Omega = \Omega^- \cup \gamma_0 \cup \Omega^+} f(x) dx + \int_{\gamma_0} f(x) dx + \int_{\Omega^+} f(x) dx = \\ &\quad \begin{matrix} \Omega = \Omega^- \cup \gamma_0 \cup \Omega^+ \\ \Omega^- \cap \gamma_0 = \{\emptyset\} \\ \Omega^+ \cap \gamma_0 = \{\emptyset\} \\ \Omega^- \cap \Omega^+ = \{\emptyset\} \end{matrix} \quad (4.15) \\ &= \int_{\Omega^-} f(x) dx + \int_{\Omega^+} f(x) dx \Leftarrow \int_{\gamma_0} 1 dx = 0 \end{aligned}$$

Para una función  $f$  cualquiera.

Así se puede concluir que a efectos de cálculo de integrales incluir el contorno es irrelevante. En adelante se considerara el contorno incluido en la región interior.

Las funciones  $\chi^{\pm}$  son funciones de la variable multidimensional  $x \in \Omega$ . A veces es mas practico trabajar con funciones de la variable unidimensional  $\phi$ . Se define como función salto o Heaviside unidimensional a

$$H(\phi) = \begin{cases} 0 & \phi \leq 0 \\ 1 & \phi > 0 \end{cases} \quad (4.16)$$

Donde la función  $\phi$  depende de  $x \in \Omega$ , aunque no es necesario especificar esta dependencia cuando se trabaja con  $H$ . Introducida esta notación podemos trabajar en el espacio unidimensional y podemos rescribir (4.12) y (4.13) como sigue:

$$\begin{aligned} \chi^+(x) &= H(\phi(x)) \\ \chi^-(x) &= 1 - H(\phi(x)) \end{aligned} \quad (4.17)$$

La integral de volumen (área o línea en 2-D o 1-D respectivamente) de una función  $f$  sobre la región interior  $\Omega^-$  es define como:

$$\int_{\Omega} f(x) \chi^-(x) dx \quad (4.18)$$

Donde el dominio de integración es todo  $\Omega$ , dado que  $\chi^-$  se idénticamente nulo en al región exterior  $\Omega^+$ . Podemos usar la función Heaviside unidimensional podemos rescribir la integral anterior

$$\int_{\Omega} f(x) (1 - H(\phi(x))) dx \quad (4.19)$$

Análogamente para la región exterior.

$$\int_{\Omega} f(x) H(\phi(x)) dx \quad (4.20)$$

Por definición, la derivada direccional de la función Heaviside  $H$  en la dirección  $\vec{N}$  normal al frente es la función Delta de Dirac

$$\hat{\delta}(x) = \nabla H(\phi(x)) \cdot \vec{N} \quad (4.21)$$

Que es función de la variable multidimensional  $x \in \Omega$ . La distribución anterior es únicamente distinta de cero en la interfaz,  $\gamma_0$ .

Se puede describir la ecuación (4.21) de la siguiente manera

$$\hat{\delta}(x) = H'(\phi(x)) \nabla \phi(x) \cdot \frac{\nabla \phi(x)}{\|\nabla \phi(x)\|} = H'(\phi(x)) \|\nabla \phi(x)\| \quad (4.22)$$

Utilizando la regla de la cadena, la definición (2.3) de la normal a la interfaz y el hecho que  $\nabla \phi(x) \cdot \nabla \phi(x) = \|\nabla \phi(x)\|^2$ .

En el caso unidimensional, la función delta se define como la derivada de la función unidimensional Heaviside

$$\delta(\phi) = H'(\phi) \quad (4.23)$$

Donde  $H(\phi)$  se define como (4.16). La función delta  $\delta(\phi)$  es idénticamente nula en todo el dominio excepto en  $\gamma_0$ .

Con la función delta podemos plantear

$$\hat{\delta}(x) = \delta(\phi(x)) \|\nabla \phi(x)\| \quad (4.24)$$

La integral de superficie (línea o punto en 2-D o 1-D respectivamente) de una función  $f$  sobre la interfaz,  $\gamma_0$ , se define como

$$\int_{\Omega} f(x) \hat{\delta}(x) dx \quad (4.25)$$

Donde la región de integración es todo el dominio  $\Omega$  dado que  $\hat{\delta}(x)$  es cero excepto en la interfaz. Utilizando (4.24) en (4.25) tenemos

$$\int_{\Omega} f(x) \delta(\phi(x)) \|\nabla \phi(x)\| dx \quad (4.26)$$

Observando (4.19), (4.20) y (4.26) se puede concluir que el calculo de integrales de volumen o superficie no necesita de algoritmos de identificación de la región interior, exterior o contorno, simplemente se integra sobre todo el dominio  $\Omega$  y los valores de la función Level Set  $\phi$  definen donde el integrando es nulo.

Se define como función Heaviside suavizada a una función definida según [18]

$$H_{\varepsilon}(\phi) = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2} \left[ 1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\pi \phi}{\varepsilon}\right) \right] & |\phi| < \varepsilon \\ 1 & \phi > \varepsilon \end{cases} \quad (4.27)$$

Esta función depende de un parámetro ajustable  $\varepsilon$  que determina el tamaño del ancho de banda para la transición entre los valores 0 y 1.

Tomado  $\varepsilon = 1.5 \cdot \ell_{\min}$ , con  $\ell_{\min}$  un tamaño mínimo característico de elemento de la malla, tenemos que el ancho de banda será 3 veces el tamaño mínimo  $\ell_{\min}$  (aproximadamente tres elementos, asumiendo que los elementos mas pequeños son los adyacentes a la interfaz) siempre que la función Level Set cumpla (2.18).

Numéricamente, el uso de la función Heaviside suavizada  $H_{\varepsilon}(\phi)$  es más adecuado que el uso de la función Heaviside  $H(\phi)$  para el calculo de integrales como (4.19) y (4.20). El

hecho de utilizar una transición suave pondera la pérdida de volumen asociada a que solo conocemos los valores de  $\phi$  en los nodos de la malla.

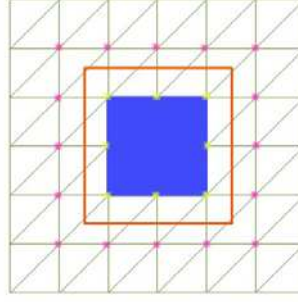


figura 9

En la figura 9 se muestra un ejemplo en que el uso de una función Heaviside sin suavizar es incorrecto.

- Líneas verdes: Definen los elementos triangulares de la malla.
- Línea roja: La posición de la interfaz.
- Nodos amarillos: Nodos con valores de la función Level Set negativos.
- Nodos violetas: Nodos con valores de la función Level Set positivos.

Si calculamos el área de la región interior de la interfaz según (4.19) tenemos

$$\mathcal{A}(\Omega^-) = \int_{\Omega} 1 - H(\phi) \quad (4.28)$$

La función  $1 - H(\phi)$  únicamente toma valores distintos de cero en la región coloreada de azul, que evidentemente tiene un área menor que la región interior de la interfaz.

En cambio el uso de una función Heaviside suavizada pondera la pérdida de área debida a la discretización numérica.

Así por ejemplo el cálculo de volúmenes necesarios para el algoritmo de conservación de masa descrito en la sección 4.1 se calculan como sigue:

$$\begin{aligned} V_{\Omega}^{n+1} &= V\left(\left\{x \in \Omega \mid \phi(x, t^{n+1}) < 0\right\}\right) = V(\Omega_{n+1}^-) = \\ &= \int_{\Omega} 1 - H_{\varepsilon}(\phi) \end{aligned} \quad (4.29)$$

## 5. Ecuaciones de Hamilton-Jacobi

En esta sección se hace una revisión general de las principales características de las ecuaciones de Hamilton-Jacobi [1,12].

Se llama ecuación de Hamilton-Jacobi a una EDP de la siguiente forma:

$$\frac{\partial \phi}{\partial t} + H(\nabla \phi) = 0 \quad (5.1)$$

Donde  $H$  puede ser función del tiempo y el espacio.

Considerando

$$H(\nabla \phi) = u \cdot \nabla \phi \quad (5.2)$$

La ecuación (5.1) se transforma trivialmente en la ecuación de transporte (2.5).

Las ecuaciones de Hamilton-Jacobi depende en la mayoría de casos de las primeras derivadas de  $\phi$  y por lo tanto se tratan de ecuaciones hiperbólicas.

### 5.1 Conexión con leyes conservativas.

Consideremos la ley de conservación escalar unidimensional.

$$\frac{\partial v}{\partial t} + \frac{\partial F(v)}{\partial x} = 0 \quad (5.3)$$

Donde  $v$  es la cantidad conservada y  $F(v)$  es la función de flujo.

Una ley de conservación bien conocida es la ecuación de continuidad

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} = 0 \quad (5.4)$$

Que impone la conservación de masa en el dominio.

Es conocido que las ecuaciones del tipo (5.3) admiten soluciones discontinuas y sus soluciones pueden llegar a desarrollar ondas de choque,...Un ejemplo estudiado en profundidad de ley de conservación es la ecuación de Burgers

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x} \left( \frac{v^2}{2} \right) = 0 \quad (5.5)$$

De la que se conoce que admite soluciones discontinuas (ondas de choque) aun cuando la condición inicial es regular.

A continuación se establece una relación entre las ecuaciones Hamilton-Jacobi y las leyes de conservación, que permitirá obtener resultados cualitativos sobre las soluciones de (5.1) [1,12].

La ecuación unidimensional de Hamilton-Jacobi aplicada a  $\phi$  es

$$\frac{\partial \phi}{\partial t} + H \left( \frac{\partial \phi}{\partial x} \right) = 0 \quad (5.6)$$

Si ahora se calcula la derivada espacial de (5.6), asumiendo suficiente regularidad (Existencia de segundas derivadas continuas) por el teorema de Schwartz o de las derivadas cruzadas:

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial x} \left( H \left( \frac{\partial \phi}{\partial x} \right) \right) &= 0 \\ (\phi_x)_t + (H(\phi_x))_x &= 0 \end{aligned} \quad (5.7)$$

Tomando  $v = \phi_x$  en la ecuación (5.7) se obtiene:

$$\frac{\partial v}{\partial t} + \frac{\partial H(v)}{\partial x} = 0 \quad (5.8)$$

Que es una ley de conservación escalar unidimensional, ecuación (5.3) tomando  $F = H$ . En vista de (5.8) se puede concluir que:

En el caso unidimensional se puede establecer una relación biunívoca entre las ecuaciones de Hamilton-Jacobi y sus leyes de conservación asociadas ( $F = H$ ).

La derivada espacial de la solución  $v$  de una ley de conservación es solución de la ecuación de Hamilton-Jacobi asociada.

A su vez la solución de una ecuación de Hamilton-Jacobi es la integral de la solución de la ley de conservación asociada.

Este hecho permite obtener información cualitativa de las soluciones de las ecuaciones de Hamilton-Jacobi.

Como la integral de una función discontinua produce una función cúspide o quebrada (con primera derivada discontinua) las soluciones de una ecuación de Hamilton-Jacobi puede desarrollar cúspides aún cuando la condición inicial sea suave.

Adicionalmente las soluciones de una ecuación de Hamilton-Jacobi no pueden desarrollar una discontinuidad a menos que la correspondiente ley de conservación desarrolle un delta de Dirac.

Por ultimo, como las leyes de conservación no admiten solución única, condiciones de entropía adicionales son necesarias para obtener una solución “físicamente” aceptable.

## 6. Implementación numérica

La implementación numérica de algunas de las ideas anteriores se ha realizado utilizando como plataforma de desarrollo el programa Tdyn.

Tdyn es un entorno de simulación de dinámica de fluidos (CFD) basado en el Método de los Elementos Finitos estabilizado, capaz de reproducir distintos tipos de turbulencias y con sofisticadas herramientas para simulación de problemas de transferencia de calor en sólidos y fluidos, problemas de transporte de sustancias como problemas de superficie libre, entre otros.

Módulos en Tdyn:

- **RANSOL:** Fluidos reales incompresibles y ligeramente compresibles, viscosos y con turbulencia.
- **HEATTRANS:** Análisis de transferencia de calor en sólidos y fluidos.
- **ADVECT:** Simulación del transporte / dispersión de múltiples sustancias considerando el efecto de los cambios las propiedades físicas como superficie libre, precipitación, etc.
- **USER SOLVER:** Integración de la ecuación de convección-difusión-reacción-producción con parámetros definidos por el usuario.
- **NAVAL:** Problemas de simulación naval.
- **PREPOST:** Entorno de pre y post proceso totalmente integrado basado en el sistema GiD.

El módulo USER SOLVER dota a Tdyn con la capacidad de resolver un amplio rango problemas físicos, en particular, el problema de interés de este estudio: la integración numérica de la ecuación de transporte o equivalentemente la ecuación de convección transitoria. Como se ha expuesto anteriormente esta ecuación es la que gobierna la evolución de la interfaz.

La ecuación convección-difusión-reacción-producción es la siguiente:

$$\underbrace{f_{trans}^1 \frac{\partial \phi}{\partial t} + f_{conv}^1 u \cdot \nabla \phi}_{conectivo} + \underbrace{\nabla \cdot (f^2 \nabla \phi)}_{difusivo} + \underbrace{f^3 \phi}_{reactivo} = \underbrace{f^4}_{productivo} \quad (6.1)$$

Donde  $f_{trans}^1, f_{conv}^1, f^2, f^3, f^4$  se pueden definir como constantes o funciones de las variables involucradas en el problema, como  $\phi, u$ , siendo  $u$  la velocidad del fluido.



Asumiendo  $f_{trans}^1 = f_{conv}^1 = 1, f^2 = f^3 = f^4 = 0$  la ecuación (6.1) se transforma en la ecuación (2.5), que junto con sus condiciones iniciales y de contorno plantea el mismo problema que (2.9).

Así el programa Tdyn ofrece la posibilidad de poder integrar la ecuación de transporte con un campo de velocidades impuesto u obtenido de la resolución de las ecuaciones promediadas de Reynolds-Navier-Stokes.

El esquema de integración de la ecuación del transporte existente en Tdyn ha sido el punto inicial para la implementación de nuevos algoritmos numéricos basados en las ideas expuesta en los capítulos anteriores.

Se presenta un objetivo más en este estudio consistente en mejorar la solución al problema (2.9) que Tdyn ofrece actualmente.

## 6.1 Esquema integración numérica ecuación de transporte.

Tdyn utiliza un esquema de integración numérica basado en el método de los elementos finitos estabilizado para la ecuación de transporte. El esquema de integración estabilizado es conocido como Calculo Finitesimal [10,14]. Este esquema modifica la ecuación de transporte original para obtener una nueva ecuación con mejores propiedades numéricas (estabilidad).

La ecuación de transporte original es:

$$\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = 0$$

Esta ecuación es modificada con un término difusivo adicional con el objetivo de poder corregir la subdifusividad asociada a la integración de la ecuación por el método de Galerkin clásico en problemas de convección dominante. La ecuación anterior es el caso limite de una ecuación elíptica donde el termino difusivo es nulo, y por consiguiente con numero de Péclet infinito.

### 6.1.1 Formulación de la ecuación de transporte utilizando Calculo Finitesimal

El Calculo Finitesimal esta basado en el balance de flujos en un dominio de tamaño finito. A continuación se presenta una breve descripción este método:

Considera el problema de convección pura estacionario en un dominio  $\Omega$  unidimensional de longitud  $L$ . La ecuación de balance de flujos en un subdominio, de  $\Omega$ , de longitud  $d$ , es (véase figura 10):

$$q_A - q_B = 0 \quad (6.2)$$

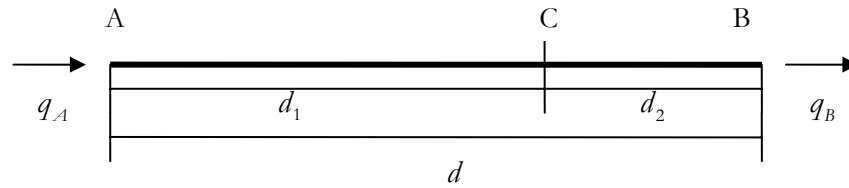


figura 10

Donde  $q_A$  y  $q_B$  son respectivamente los flujos conectivos entrante y saliente, respectivamente.

En el caso de un problema de convección puro, como es la ecuación de transporte, el flujo es expresa como  $q = -u\phi$ .

Podemos, usando series de Taylor, expresar el valor de los flujos en A y B en función del flujo en un punto interior C arbitrario en un subdominio de longitud  $d$ .

$$\begin{aligned} q_A &= q_C - d_1 \left. \frac{dq}{dx} \right|_C + \frac{d_1^2}{2} \left. \frac{d^2q}{dx^2} \right|_C - O(d_1^3) \\ q_B &= q_C + d_2 \left. \frac{dq}{dx} \right|_C + \frac{d_2^2}{2} \left. \frac{d^2q}{dx^2} \right|_C - O(d_2^3) \end{aligned} \quad (6.3)$$

Substituyendo (6.3) en (6.2) obtenemos:

$$\frac{dq}{dx} - \frac{b}{2} \frac{d^2q}{dx^2} = 0 \quad (6.4)$$

Donde  $b = d_1 - d_2$  y todas las derivadas son calculadas en el punto C. Utilizando el calculo infinitesimal clásico, la ecuación de balance de flujos se considera en un domino infinitesimal, y tendríamos que (6.4) resultaría  $\frac{dq}{dx}$ .

El término introducido usando Calculo Finitesimal se puede interpretar como una difusividad artificial que tiene un efecto estabilizador.

El caso general (transitorio) la formulación basada en el Calculo Finitesimal de la ecuación de transporte es:

$$r - \frac{1}{2} b^t \nabla r - \frac{1}{2} \lambda \frac{\partial r}{\partial t} = 0 \quad (6.5)$$

Con

$$r = \frac{\partial \phi}{\partial t} + u \cdot \nabla \phi \quad (6.6)$$

En la ecuación (6.5),  $b = (b_x, b_y, b_z)^t$  (para el problema 3-D) contiene los términos de estabilización espacial y  $\lambda$  el termino de estabilización temporal.

El vector  $b$  se puede descomponer como:

$$b = b_s \frac{u}{\|u\|} + b_t \frac{\nabla \phi}{\|\nabla \phi\|} \quad (6.7)$$

Además,  $b_s, b_t$  pueden, para cada elemento  $e$ , ser expresados en función de la dimensión característica del elemento  $l^{(e)}$ .

$$b_s^{(e)} = \alpha_s^{(e)} l^{(e)} \quad b_t^{(e)} = \alpha_t^{(e)} l^{(e)} \quad (6.8)$$

Tomando  $\alpha_t^{(e)} = 0$  recuperamos la estabilización introducida por el método SUPG.

Una versión más simplificada de la ecuación (6.5) se obtiene considerando únicamente los términos adicionales en segundas derivadas espaciales

$$\frac{\partial \phi}{\partial t} + (u \cdot \nabla) \phi - \frac{1}{2} b^t \nabla \cdot ((u \cdot \nabla) \phi) = 0 \quad (6.9)$$

Esta versión estabilizada de la ecuación de transporte es la utilizada en este estudio para su integración numérica.

El hecho de despreciar el término de estabilización temporal obliga a utilizar un esquema de integración temporal estable.

Dado que la ecuación de transporte es una ecuación de convección pura el término de estabilización transversal,  $b_t$  se puede considerar nulo.

El vector longitud característica  $b$  se puede expresar entonces como:

$$b = \tau u \quad (6.10)$$

Donde el parámetro de estabilización numérica  $\tau$ , llamado tiempo intrínseco, se calcula de forma heurística como:

$$\tau = \begin{cases} \frac{|u_x^{elem}| + |u_y^{elem}| + |u_z^{elem}|}{2\|u\|^2} & \|u\| < Min\_tol \\ \frac{\Delta t}{2} & \|u\| > Min\_tol \end{cases} \quad (6.11)$$

$$u = (u_x, u_y, u_z) \quad 0 < Min\_tol \ll 1$$

$$l_s^{elem} = \max \{ |s_i - s_j| \mid i, j = 1, \dots, n_e \quad s = x, y, z \}$$

Donde  $n_e$  es el número de nodos por elemento.

### 6.1.2 Integración Temporal

Para resolver numéricamente la ecuación (6.12) utilizaremos el esquema implícito backward Euler, que es incondicionalmente estable.

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = - (u^{n+1} \cdot \nabla) \phi^{n+1} + \frac{1}{2} \tau^{n+1} (u^{n+1})^t \nabla \cdot ((u^{n+1} \cdot \nabla) \phi^{n+1}) \quad (6.13)$$

Donde se ha hecho uso de la ecuación (6.10) y el superíndice denota el paso de tiempo en que se evalúan las cantidades.

El método backward Euler es de primer orden en el tiempo e introduce amortiguamiento notable en todos los armónicos de la solución. Este hecho puede ser interesante en el caso que la solución presente grandes oscilaciones.

### 6.1.3 Discretización Espacial

La discretización espacial se realiza mediante el método estándar de Galerkin aplicado a la ecuación (6.13) [23].

Primero ordenamos los términos de la ecuación (6.13). En el lado izquierdo, los términos evaluados en el instante de tiempo  $n+1$  y en el lado derecho, los términos evaluados en el instante de tiempo  $n$

$$\underbrace{\frac{\phi^{n+1}}{\Delta t}}_{\text{temporal}} + \underbrace{(u^{n+1} \cdot \nabla) \phi^{n+1}}_{\text{convectivo}} - \underbrace{\frac{1}{2} \tau^{n+1} (u^{n+1})^t \nabla \cdot ((u^{n+1} \cdot \nabla) \phi^{n+1})}_{\text{estabilizador}} = \underbrace{\frac{\phi^n}{\Delta t}}_{\text{temporal}} \quad (6.14)$$

Buscamos una solución débil de (6.14) de la forma:

$$\tilde{\phi}(x, t) = \sum_j^{\text{totalnodes}} \phi_j N_j(x, t) \quad (6.15)$$

Donde  $N_j$  son funciones de forma globales,  $\phi_j$  son los valores nodales y  $totalnodes$  numero total de nodos de la discretización.

Sea  $\Omega^{(e)}$  el domino asociada a un elemento finito  $e$ . Se define como función de forma local a

$$N_j^{(e)}(x, t) = N_j(x, t)|_{\Omega^{(e)}} \quad J = 1 \div nnodes \quad (6.16)$$

Donde  $J = CN(j, e)$  con  $CN$  la matriz de conectividades y  $nnodes$  el numero de nodos del elemento  $e$

Sea  $v$  una función test, consideramos la forma débil de la ecuación (6.14)

$$\frac{1}{\Delta t} \int_{\Omega} v \phi^{n+1} + \int_{\Omega} v (u^{n+1} \cdot \nabla) \phi^{n+1} - \frac{1}{2} \int_{\Omega} v \tau^{n+1} (u^{n+1})^t \nabla \cdot ((u^{n+1} \cdot \nabla) \phi^{n+1}) = \frac{1}{\Delta t} \int_{\Omega} v \phi^n \quad (6.17)$$

En adelante, para comodidad en la notación, se omitirá el superíndice temporal de las variables el término convectivo y estabilizador.

Consideramos por separado los sumandos de la ecuación (6.17):

**-Término convectivo:**

Tomamos como función test  $v = N_i$  y considerando que  $u \simeq \tilde{u} = \sum_k N_k u_k$

$$\int_{\Omega} v(u \cdot \nabla) \phi \simeq \int_{\Omega} N_i (N_k u_k \cdot \nabla N_j) \phi_j d\Omega = \sum_e^{nelem} \int_{\Omega^{(e)}} N_i^{(e)} (N_k^{(e)} u_k^{(e)} \cdot \nabla N_j^{(e)}) \phi_j^{(e)} = C_{ij} \phi_j \quad (6.18)$$

En (6.18) se utiliza la notación de Einstein para índices repetidos.

$$a_k b_k = \sum_k a_k b_k \quad (6.19)$$

$$C_{ij} = \sum_e^{nelem} C_{ij}^{(e)} = \sum_e^{nelem} C_{CN(I,e)CN(J,e)}^{(e)} = \mathbb{A}_e^{nelem} C_{IJ}^e \quad (6.20)$$

Donde  $\mathbb{A}$  denota el proceso de ensamblaje

$$M_{nnodes \times nnodes}(\mathbb{R}) \xrightarrow{\mathbb{A}} M_{totalnodes \times totalnodes}(\mathbb{R}) \quad (6.21)$$

$$C_{IJ}^{(e)} \quad C_{ij}^{(e)}$$

La matriz  $C_{IJ}^{(e)}$  se llama matriz elemental asociada al elemento  $e$

**- Término de estabilización:**

$$\int_{\Omega} v \tau u' \nabla ((u \cdot \nabla) \phi) d\Omega = \int_{\Omega} \tau v (u \cdot \nabla) ((u \cdot \nabla) \phi) d\Omega \quad (6.22)$$

Antes de discretizar (6.22) operamos el integrando de este término aplicando la regla de derivación de un producto:

$$\begin{aligned} \tau(u \cdot \nabla) [v \cdot (u \cdot \nabla) \phi] &= \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] + \tau v \cdot (u \cdot \nabla) [(u \cdot \nabla) \phi] = \\ &= \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] + v \tau u' \nabla ((u \cdot \nabla) \phi) \end{aligned} \quad (6.23)$$

Combinado (6.23) en (6.22) y usando el Teorema de la divergencia:

$$\begin{aligned} \int_{\Omega} \tau v (u \cdot \nabla) ((u \cdot \nabla) \phi) d\Omega &= \int_{\Omega} \tau (u \cdot \nabla) [v (u \cdot \nabla) \phi] d\Omega - \int_{\Omega} \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] d\Omega = \\ &= \int_{\partial\Omega} \tau v (u \cdot \nabla) \frac{\partial \phi}{\partial n} d\Gamma - \int_{\Omega} \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] d\Omega \end{aligned} \quad (6.24)$$

Asumiendo que  $\frac{\partial \phi}{\partial n} = 0$  en  $\partial\Omega$ , (6.24) se reduce a

$$\int_{\Omega} \tau v (u \cdot \nabla) ((u \cdot \nabla) \phi) d\Omega = - \int_{\Omega} \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] d\Omega \quad (6.25)$$

Ahora discretizamos el lado derecho de (6.25) que es equivalente al término de estabilización.

Tomamos como función test  $v = N_i$  y considerando que  $u \simeq \tilde{u} = \sum_k N_k u_k$

$$\begin{aligned}
\int_{\Omega} \tau [(u \cdot \nabla) v] [(u \cdot \nabla) \phi] d\Omega &= \int_{\Omega} \tau (N_k u_k \cdot \nabla N_j) N_i (N_k u_k \cdot \nabla N_l) \phi_l = \\
&= \sum_e^{nelem} \int_{\Omega^{(e)}} \tau^{(e)} (N_k^{(e)} u_k^{(e)} \cdot \nabla N_j^{(e)}) N_i^{(e)} (N_k^{(e)} u_k^{(e)} \cdot \nabla N_l^{(e)}) \phi_l^{(e)} = S_{ij} \phi_j
\end{aligned} \tag{6.26}$$

El proceso de ensamblaje de  $S_{ij}$  es análogo al ensamblaje del término conectivo.

### -Término temporal

Tomamos como función test  $v = N_i$ .

$$\frac{1}{\Delta t} \int_{\Omega} v \phi = \frac{1}{\Delta t} \int_{\Omega} N_i N_j \phi_j = \frac{1}{\Delta t} M_{ij} \phi_j \tag{6.27}$$

Es habitual, con el fin de simplificar los cálculos, colapsar la matriz  $M$  en forma diagonal (lumped mass matrix). El proceso de diagonalización se realiza de la siguiente forma:

$$L_{ij} = \begin{cases} 0 & i \neq j \\ \sum_j M_{ij} & i = j \end{cases} \tag{6.28}$$

Se puede comprobar que  $L$  es una buena aproximación de  $M$ , y que en caso de usar una regla de integración numérica nodal para (6.27) es exacta.

Se entiende por regla de integración nodal, aquella que los puntos de integración coinciden con nodos.

#### 6.1.3.1 Expresiones indiciales

En esta sección se presentan las ecuaciones (6.18), (6.26) y (6.27) en su forma indicial completa para las matrices elementales. Consideramos un elemento finito fijo  $e$ .

Introducimos la siguiente notación:

- $nnode$  : Numero de nodos elemento  $e$
- $n \dim$  : Dimensión del espacio
- $n \times m = M_{n \times m}(\mathbb{R})$

Dado que el elemento finito  $e$  es fijo, en adelante se omite el superíndice ( $e$ ) en la notación.

#### Término convectivo:

$$\int_{\Omega} N_i \left( N_k u_{k,n} \frac{\partial N_j}{\partial x_n} \right) \phi_j d\Omega = \left( \int_{\Omega} \underbrace{N_i N_k}_{nnode \times nnode} \cdot \underbrace{u_{k,n}}_{nnode \times n \dim} \cdot \underbrace{\frac{\partial N_j}{\partial x_n}}_{n \dim \times nnode} d\Omega \right) \underbrace{\phi_j}_{nnode \times 1} \tag{6.29}$$

#### Término de estabilización:

$$\begin{aligned}
&\int_{\Omega} \left( N_k u_{k,n} \frac{\partial N_j}{\partial x_n} \right) N_i \left( N_k u_{k,n} \frac{\partial N_l}{\partial x_n} \right) \phi_l d\Omega = \\
&= \int_{\Omega} \left( N_k u_{k,n} \frac{\partial N_l}{\partial x_n} N_i \right)^t \left( N_k u_{k,n} \frac{\partial N_l}{\partial x_n} \right) \phi_l d\Omega = \\
&= \left( \int_{\Omega} \underbrace{u_{k,n} \frac{\partial N_l}{\partial x_n}}_{nnode \times nnode} \cdot \underbrace{\left( N_k \right)^t}_{1 \times nnode} \cdot \underbrace{N_k u_{k,n} \frac{\partial N_l}{\partial x_n}}_{nnode \times nnode} d\Omega \right) \underbrace{\phi_l}_{nnode \times 1}
\end{aligned} \tag{6.30}$$

En (6.30) se ha utilizado las siguientes propiedades:

$$(AB)^t = B^t A^t \quad (6.31)$$

$$N_k u_{k,n} \frac{\partial N_l}{\partial x_n} N_l^i \rightarrow (1 \times nnode)(nnode \times n \dim)(n \dim \times nnode)(nnode \times 1) \quad (6.32)$$

$N^i$  es el vector de valores nodales de la función de forma del nodo  $i$ , (0 excepto en la posición  $i$ ). Entonces

$$N_l^i = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} = Id_{nnode \times nnode} \quad (6.33)$$

### Término temporal

$$\frac{1}{\Delta t} \int_{\Omega} N_i N_j \phi_j = \frac{1}{\Delta t} \left( \int_{\Omega} \underbrace{N_i N_j}_{nnode \times nnode} \right) \phi_j \quad (6.34)$$

#### 6.1.4 Ecuación de transporte discretizada por Calculo Finitesimal

Utilizando (6.18), (6.26) y (6.27), obtenemos la discretización por el método de los Elementos Finitos estabilizados de la ecuación de transporte

$$\left[ \frac{1}{\Delta t} \int_{\Omega} N_i N_j d\Omega + \int_{\Omega} N_i (N_k u_k^{n+1} \cdot \nabla N_j) d\Omega + \int_{\Omega} \tau (N_k u_k^{n+1} \cdot \nabla N_l) N_i (N_k u_k^{n+1} \cdot \nabla N_j) d\Omega \right] \phi_j^{n+1} = \left[ \frac{1}{\Delta t} \int_{\Omega} N_i N_j d\Omega \right] \phi_j^n \quad (6.35)$$

#### 6.1.5 Comentarios sobre la ecuación de reinicialización

Como se vio en la sección 2.1.3 es necesaria la integración de la ecuación (2.26) para reinicializar la función Level Set y conservar sus propiedades de función distancia con signo.

$$\frac{\partial d}{\partial \tau} + \omega \cdot \nabla d = \text{sign}(\phi)$$

Esta ecuación tiene los mismo términos que la ecuación del transporte (término temporal y término convectivo). El lado derecho de la ecuación, término de reacción, no influye en el método de estabilización y puede ser discretizado por el método de Galerkin clásico de forma análoga al término temporal.

El único aspecto nuevo de esta ecuación que merece una especial atención es

$$\omega = \text{sign}(\phi) \frac{\nabla d}{\|\nabla d\|}. \text{ La "velocidad" } \omega \text{ requiere el calculo del gradiente de la función } d.$$

Calcular el gradiente de función es equivalente a resolver la siguiente EDP:

$$\nabla d = g \quad (6.36)$$

Aplicamos el método de Galerkin a (6.36).

Sea  $v$  una función test.

$$\int_{\Omega} (v \cdot \nabla) d \, d\Omega = \int_{\Omega} v g \, d\Omega \quad (6.37)$$

Considerando  $v = N_i$

$$\left( \int_{\Omega} (N_i \cdot \nabla) N_j d\Omega \right) d_j = \left( \int_{\Omega} N_i \cdot \nabla N_j d\Omega \right) d_j = \left( \int_{\Omega} N_i N_j d\Omega \right) g_j \quad (6.38)$$

Aglutinando la matriz  $M$  en forma diagonal como en (6.39)

$$L_{ii} = \sum_j M_{ij} = \sum_j \int_{\Omega} N_i N_j d\Omega$$

$$[L] = \begin{pmatrix} L_{11} & & 0 \\ & \ddots & \\ 0 & & L_{nnode, nnode} \end{pmatrix} \quad (6.39)$$

$$g_j = \left( \int_{\Omega} N_i N_j d\Omega \right)^{-1} \left( \int_{\Omega} N_i \cdot \nabla N_j d\Omega \right) d_j = \frac{1}{L_{jj}} \left( \int_{\Omega} N_i \cdot \nabla N_j d\Omega \right) d_j \quad (6.40)$$

De este modo se evita resolver un sistema de ecuaciones lineales como (6.38)

## 7. Implementación algorítmica

En esta sección se presenta los algoritmos computacionales desarrollados para la implementación del Level Set Method en la plataforma TDYN. Los algoritmos se han desarrollado en lenguaje C++ y utilizando las clases y objetos existentes en TDYN.

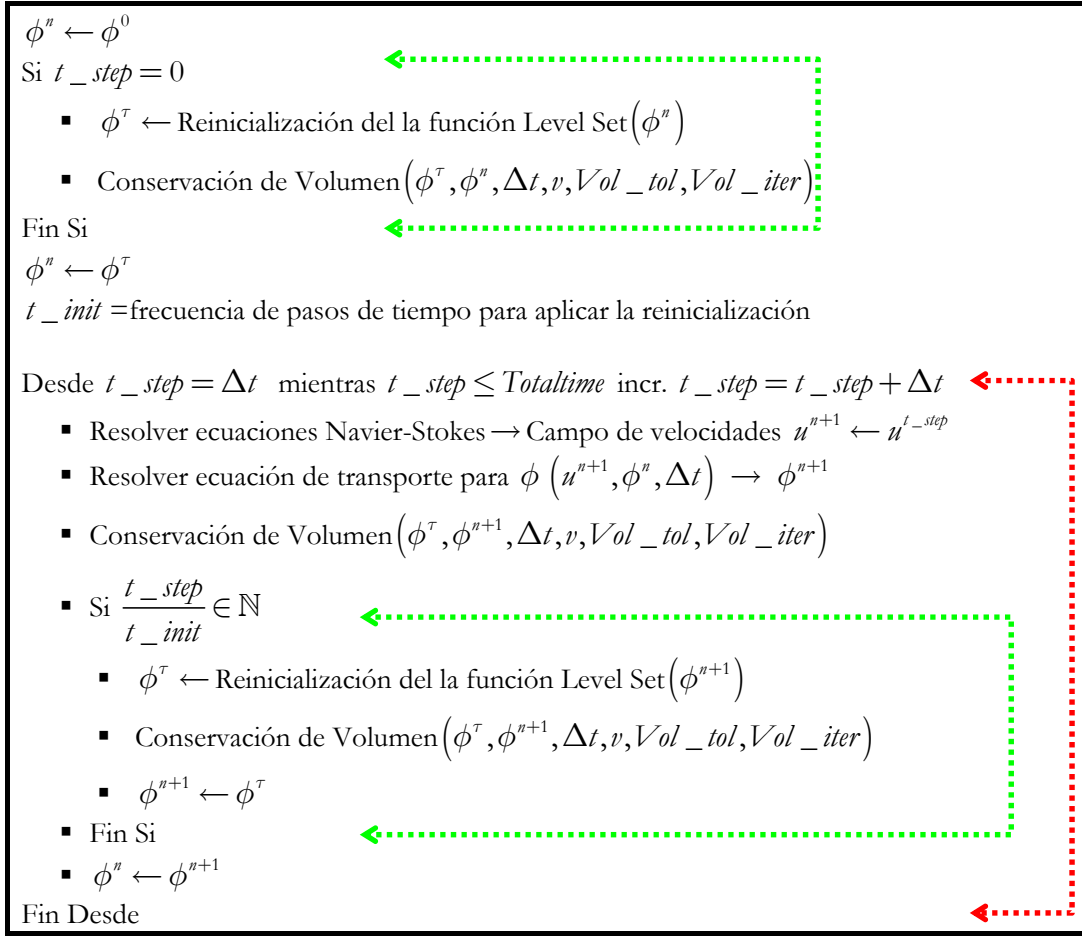
Los algoritmos se presentaran en pseudocódigo.

El Algoritmo 1 describe la implementación del método Level Set para la simulación de la evolución de la interfaz entre dos fluidos. El algoritmo se debe inicializar con los siguientes datos:

- $\phi^0$  valor inicial de la función Level Set, no es necesario que sea una función distancia con signo. Es equivalente a la posición inicial de la interfaz.
- *Totaltime* tiempo total de la simulación.
- *t\_init* frecuencia con que se aplica el algoritmo de reinicialización. Con *t\_init* = 1 es equivalente a reinicializar la función en cada paso de tiempo.
- $\Delta t$  incremento de tiempo.
- *Vol\_tol* Tolerancia sobre el error en la conservación de volumen.
- *Vol\_iter* iteraciones máximas para el algoritmo de conservación de volumen.

La subrutina “Reinicialización de la función Level Set” descrita en la sección 2.1.3 se presenta posteriormente en Algoritmo 2 y la subrutina “Conservación de Volumen” descrita en la sección 4.1 se presenta en Algoritmo 3.

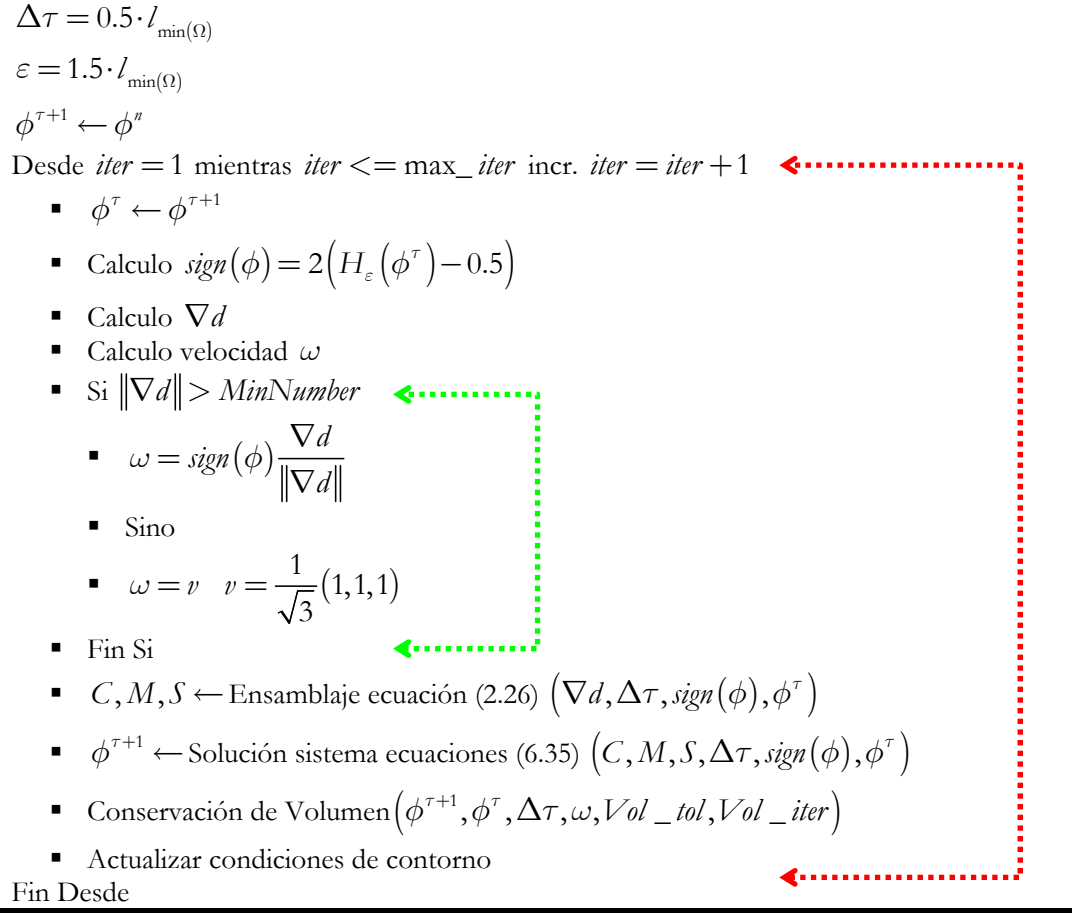
Las variables que aparecen entre paréntesis corresponden a los argumentos de las distintas subrutinas.

**Algoritmo 1**

La subrutina “Reinicialización de la función Level Set” como se vio en 2.1.3 admite dos posibles implementaciones:

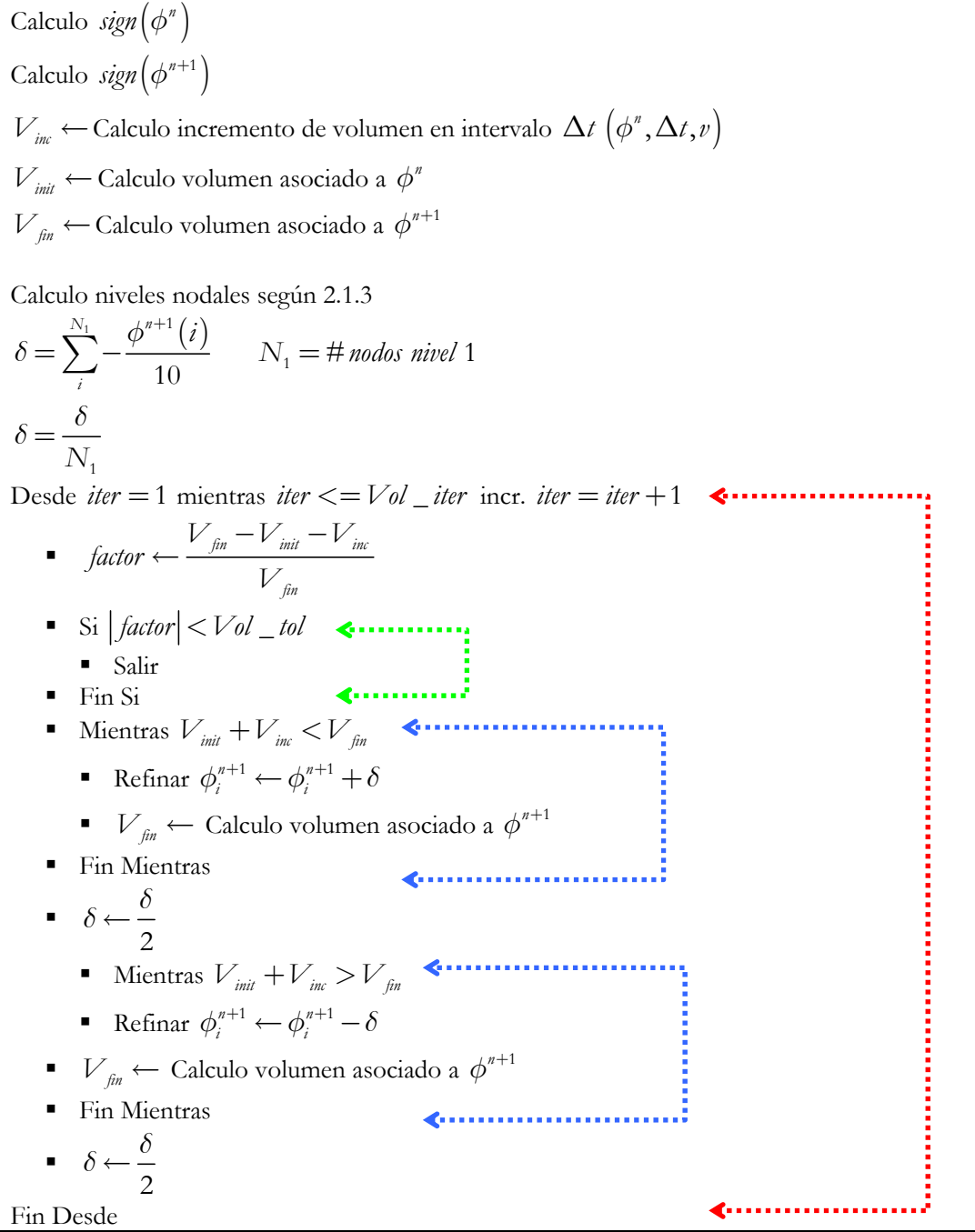
- *Algoritmo geométrico puro:* Función distancia basada en niveles nodales. Este algoritmo fue introducido con suficiente detalle en la sección 2.1.3 y no es necesaria su descripción en pseudocódigo.
- *Algoritmo matemático puro:* Integración de la ecuación (2.26). El Algoritmo 2 presenta el proceso de integración numérica de la ecuación (2.26) para la reinicialización de la función Level Set. Este algoritmo se debe inicializar con los siguientes datos:
  - $l_{\min(\Omega)}$  tamaño mínimo característico de los elementos de la discretización.
  - $\Delta\tau = 0.5 \cdot l_{\min(\Omega)}$  el paso del pseudotiempo utilizado en la integración de (2.26).
  - $\varepsilon$  parámetro que define el ancho de banda de la suavización introducida por la función Heaviside.
  - $\max\_iter$  número máximo de paso de pseudotiempo.



**Algoritmo 2**

Las variables que aparecen entre paréntesis corresponden a los argumentos de las distintas subrutinas.

La subrutina “Conservación de Volumen” que utilizan los algoritmos anteriores y descrita en 4.1 de forma teórica, se ha implementado utilizando los conceptos expuesto en 4.3 para el cálculo de integrales. En Algoritmo 3 se presenta el pseudocódigo de este algoritmo.



Algoritmo 3

## 8. Ejemplos

### 8.1 Rotación sólido rígido disco de Zalesak



figura 11

En este ejemplo se considera la rotación de sólido rígido de un disco con una hendidura como en la figura 11. La geometría de este ejemplo se describe en figura 12, todas las magnitudes están expresadas en Sistema Internacional de unidades  $(m, s)$ .

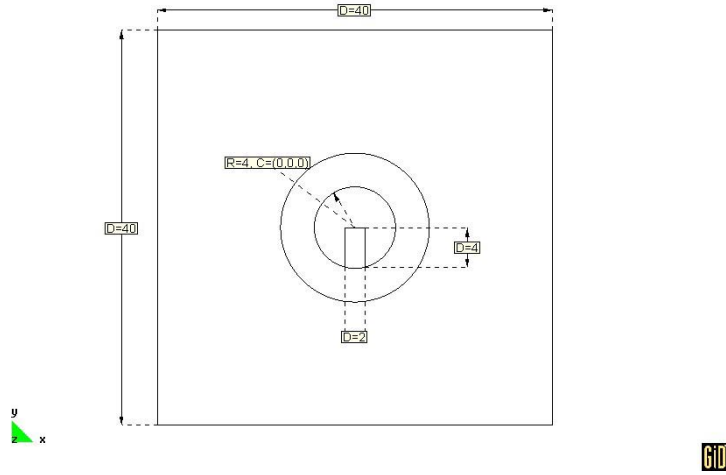


figura 12

El campo de vorticidad constante impuesto en el dominio computacional es:

$$\begin{aligned} v_x &= \frac{\pi}{2} y \\ v_y &= -\frac{\pi}{2} x \end{aligned} \tag{8.1}$$

Se deduce entonces que

$$\begin{aligned}
v &= \sqrt{v_x^2 + v_y^2} & v &= \omega r \\
v &= \sqrt{\left(\frac{\pi}{2}x\right)^2 + \left(-\frac{\pi}{2}y\right)^2} & &= \sqrt{\left(\frac{\pi}{2}r \cos \theta\right)^2 + \left(-\frac{\pi}{2}r \sin \theta\right)^2} = \frac{\pi}{2}r \text{ m} \cdot \text{s}^{-1} \quad (8.2) \\
& \quad \quad \quad \begin{matrix} x=r \cos \theta \\ y=r \sin \theta \end{matrix} \\
\omega &= \frac{\frac{\pi}{2}r}{r} = \frac{\pi}{2} \text{ rad} \cdot \text{s}^{-1}
\end{aligned}$$

Entonces para realizar una rotación completa  $2\pi \text{ rad}$  se necesitaran 4 segundos exactamente. Se han considerado incrementos de tiempo de 0.01 segundos, 400 incrementos para una rotación completa.

Los objetivos principales de este ejemplo son:

- Comprobar como el método Level Set ayuda en la conservación de la geometría del disco durante la evolución temporal del transporte.
- Verificar el algoritmo de conservación de volumen.
- Comparar los resultados basados en el método Level Set frente a la resolución de la ecuación de transporte estabilizada con una función inicial discontinua (escalón).

Para relacionar este ejemplo con la evolución de interfaces entre fluidos, se puede considera que el domino computacional esta compuesto de dos fluidos con iguales características, como en la figura 11: El fluido amarillo define el disco y el fluido verde ocupa el resto de domino.

Como el campo de velocidades impuesto es estacionario, no depende del tiempo, sabemos que la solución de la ecuación de transporte cumple:

$$\phi(x, t) = \phi_0(x - ut) \quad (8.3)$$

Donde  $\phi_0(x)$  es la posición inicial de la interfaz. Así como el campo de velocidades impuesto es una rotación en sentido contra horario, es de esperar que la evolución de la interfaz sea equivalente a la rotación de sólido rígido de la posición inicial de la interfaz, o sea del disco.

A continuación se presenta los resultados obtenidos en la simulación de la rotación del disco utilizando y sin utilizar el método Level Set, a modo de comparativa. En ambas simulaciones se ha utilizado la misma discetización espacial, elementos triangulares de 3 nodos. La malla generada ha sido de tipo no estructurado (véase figura 14) y se han asignado tamaños especiales a las líneas que define la posición inicial de la interfaz (0.1 m) así como a las superficies adyacentes (0.6 m) (véase figura 13), para el resto de domino computacional se ha utilizado un tamaño de elemento general de 2 m.

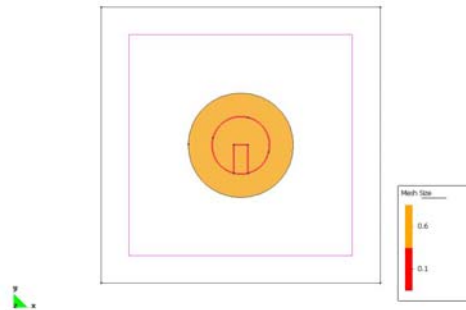


figura 13

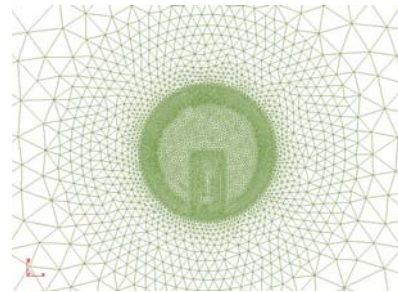


figura 14

En el caso de la simulación utilizando el método Level Set los parámetros de los distintos algoritmos descritos en la sección 7 se han ajustado con los siguientes valores:

- Frecuencia para la Reinicialización.  $t_{init}$  : 50 iteraciones (y antes del paso de tiempo inicial).
- Paso de pseudotiempo,  $\Delta\tau$  :  $0.5 \cdot l_{\min(\Omega)}$
- Numero de pasos de pseudotiempo,  $\max\_iter$  : 30
- Parámetro Función Heaviside,  $\varepsilon$  :  $1.5 \cdot l_{\min(\Omega)}$
- Iteraciones Máximas Conservación de Volumen,  $Vol\_iter$  : 5
- Tolerancia Conservación de Volumen,  $Vol\_tol$  :  $1e-5$

En tabla 1 se presenta la evolución de la interfaz obtenida sin utilizar el método Level Set y con conservación de volumen activada.

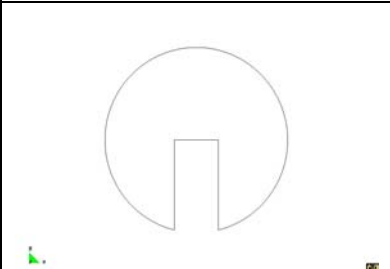
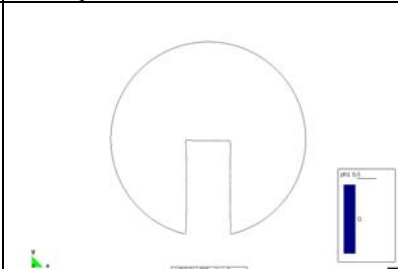
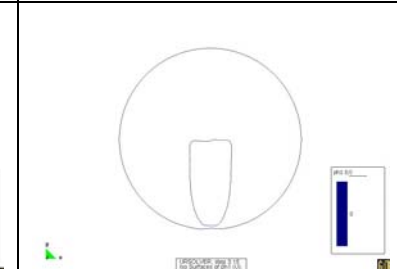
Posición Teórica	Aproximación Inicial t=0s	Una vuelta t=400s
		

tabla 1

Como se puede observar en tabla 1 aún partiendo de una aproximación inicial de la posición de la interfaz muy buena, la forma del frente después de una vuelta es dramáticamente diferente. La hendidura original se ha cerrado creando un agujero. Este error hace que la simulación de la evolución de la interfaz no confiera ninguna seguridad sobre la forma final del frente en otros casos más complejos en que la evolución del frente no sea tan evidente. Es claro que a lo largo de la simulación se ha experimentado una pérdida de volumen substancial que no ha sido corregida.

En tabla 2 se presenta la evolución de la interfaz utilizando el método Level Set e imponiendo conservación de volumen.

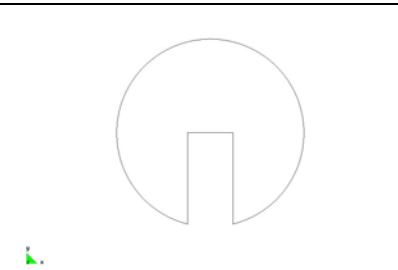
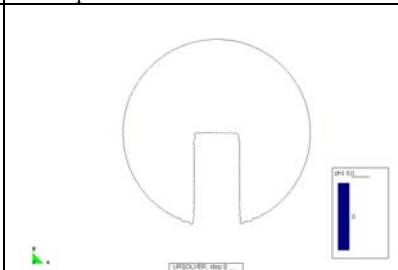
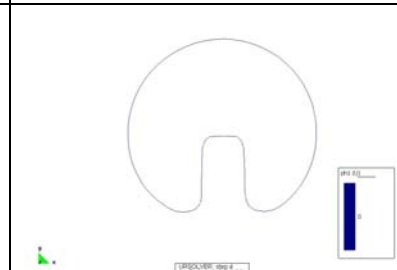
Posición Teórica	Aproximación Inicial t=0s	Una vuelta t=400s
		

tabla 2

En este caso partiendo de una aproximación inicial de calidad equivalente al caso anterior los resultados obtenidos al cabo de una rotación completa se ajustan mucho más a la realidad. Cabe destacar los siguientes aspectos de los resultados obtenidos:

- 1- Conservación de la “topología” de la interfaz: Aun cuando la forma del disco no es exactamente la esperada, en este caso el resultado obtenido respeta la forma original del frente. No se han creado conexiones ni agujeros inexistentes en la topología inicial de la interfaz. Este hecho confiere al método Level Set un grado de fiabilidad mayor que los resultados anteriores.

- 2- El volumen del disco original se ha conservado durante la evolución. Se puede concluir que el nuevo algoritmo de conservación de volumen es eficiente.
- 3- Difusividad numérica en las zonas con cambios de curvatura muy importantes, equinas: Se observa como las equinas de la hendidura se han redondeado de forma notable durante la evolución temporal. Este fenómeno es típico cuando se localiza en partes del frente difusividad numérica asociada al método de discretización utilizado.
- 4- El algoritmo de conservación de volumen potencia los efectos de la difusividad numérica: Dado que una vez realizado el balance de volumen neto perdido, se “añade” función Level Set según la normal a la interfaz produce que el redondeo de la esquinas aún se acentúe mas. Este problema se podría resolver con el diseño de algún tipo de algoritmo que localice donde se ha producido la pérdida de volumen y reintroducir función Level Set únicamente en esas zonas. Algoritmos en con este objetivo han sido planteados en [8,9,13] pero su coste computacional es extremadamente elevado comparativamente. Se plantea la siguiente cuestión: ¿Con que detalle se quiere capturar la evolución de la interfaz durante la simulación numérica? Por ejemplo durante el proceso de rellenado de un modelo o la simulación de combustión quizás la forma exacta del frente no sea absolutamente necesaria.

A continuación se presentan una sucesión de gráficos que muestran las diferencias obtenidas cuando la posición de la interfaz bien definida por una función discontinua o definida por una función distancia con signo (según el método Level Set).



figura 15

Los gráficos corresponden en todos los casos a la variación de la función Level Set la lo largo de un corte horizontal entre -4 m y 4 m con ordenada -3m. (Véase figura 15).

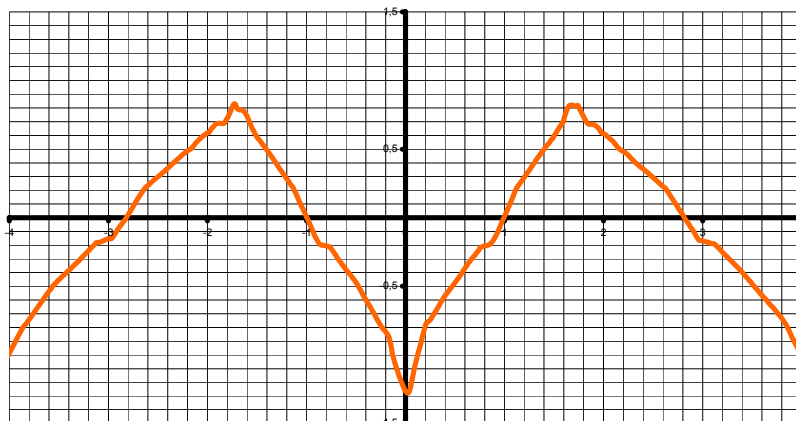


gráfico 1

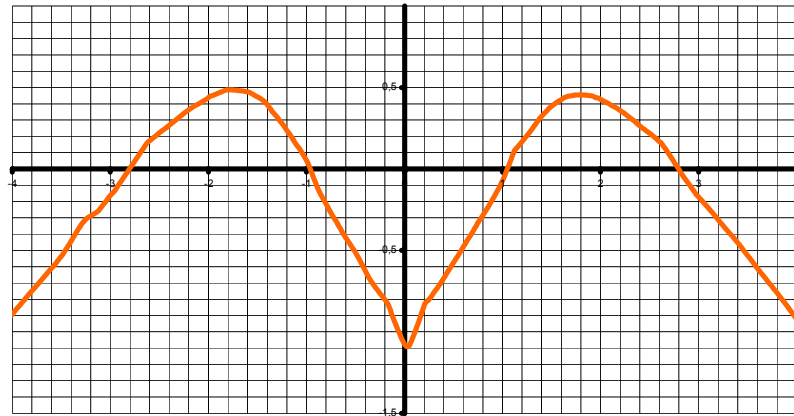


gráfico 2

Los gráfico 1 y gráfico 2 corresponden a la función distancia con signo obtenida utilizando el método Level Set. En gráfico 1 se presenta la función Level Set correspondiente a la distribución inicial ( $t=0$  seg), es decir, gráfico 1 presenta el resultado obtenido de aplicar el algoritmo de Reinicialización presentado en Algoritmo 2 a una distribución inicial como la graficada en gráfico 3.

En gráfico 2 se presenta la función Level Set después de una revolución completa ( $t=4$  seg). Como se puede observar en ambos casos la función Level Set es “suave” y efectivamente define una función distancia con signo a la interfaz.

Dado que este ejemplo es bidimensional, al realizar el corte, la restricción de la interfaz al corte es un problema unidimensional y la interfaz se reduce a puntos.

Por la teoría del método Level Set, la interfaz se recupera como la curva de nivel cero de la función Level Set. En estos casos unidimensionales, la interfaz es la intersección de la función Level Set con el eje OX. Como se ha impuesto rotación de sólido rígido, después de una rotación completa, la posición de interfaz debería ser igual a la posición inicial. En conclusión la intersección de la función Level Set con el eje OX debe coincidir en gráfico 1 y gráfico 2.

Observando los gráficos gráfico 1 gráfico 2 se puede ver que efectivamente la posición de la interfaz coincide en ambos pasos de tiempo  $t=0$  s. y  $t=400$  s. En general se puede concluir que el aspecto de la función Level Set, después de su transporte durante una vuelta completa, es bastante similar al original, como era de esperar.

Por ultimo, otro aspecto a destacar es que la integral de la función Level Set se conserva bastante bien entre ambos casos. Este hecho verifica que el algoritmo de conservación de volumen funciona correctamente. La variación de volumen obtenida es del 2% aproximadamente.

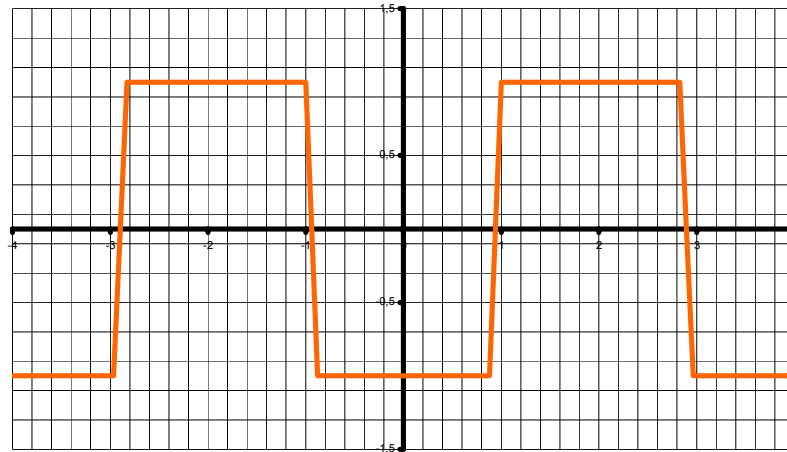


gráfico 3

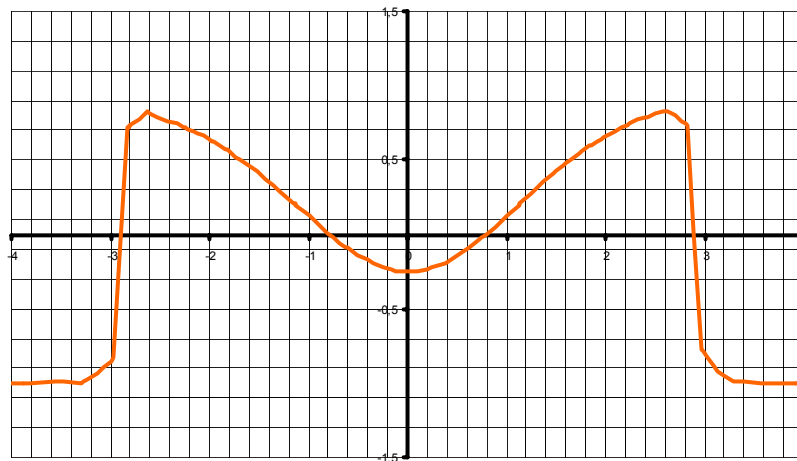


gráfico 4

Los gráfico 3 y gráfico 4 corresponden a la función escalón obtenida sin utilizar el método Level Set (sin utilizar el algoritmo de Reinicialización). En gráfico 3 se grafica la función escalón correspondiente a la distribución inicial ( $t=0$  seg), es decir, el gráfico 3 presenta una función discontinua, que toma valor 1 en el interior del disco y -1 en el exterior. En gráfico 4 se presenta la función escalón después de una revolución completa ( $t=4$  seg).

Observando los gráficos gráfico 3 y gráfico 4 se puede concluir que el transporte de una función escalón tiene grandes errores numéricos asociado. Como ya se adelantó teóricamente en secciones anteriores, se observa como en las abscisas donde se localizan los saltos discontinuos de la función Level Set (en particular, abscisas que localizan la posición de interfaz) se presenta los mayores errores numéricos asociado a la difusividad numérica inherente a la proceso de discretización, aún utilizando un esquema de integración estabilizado para la ecuación del transporte. Este error es el que produce los cambios en la topología de la interfaz.

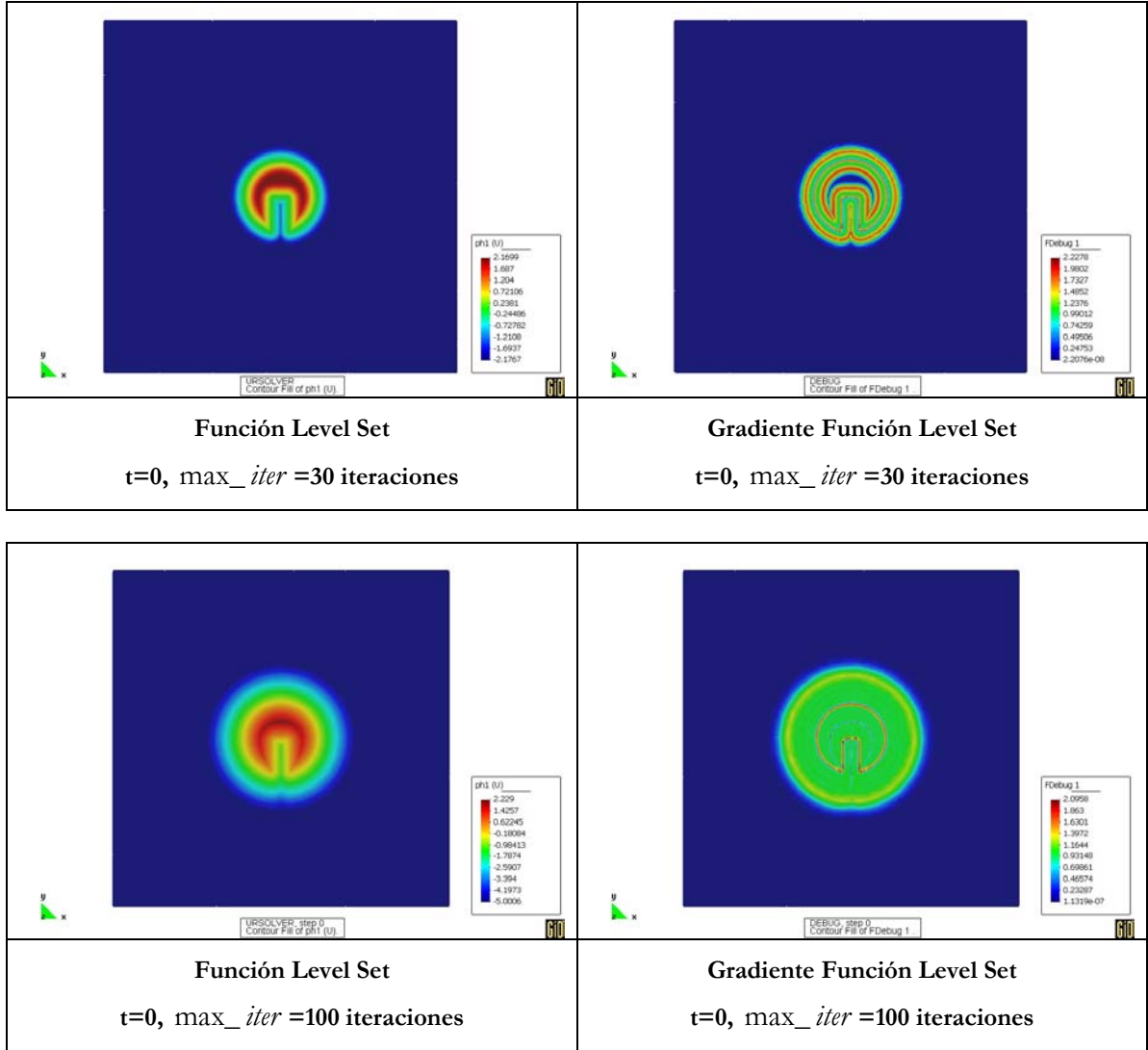
En general se observa como el aspecto inicial de función escalón se ha perdido totalmente y como la difusividad numérica ha producido un “suavizado” de la distribución inicial.

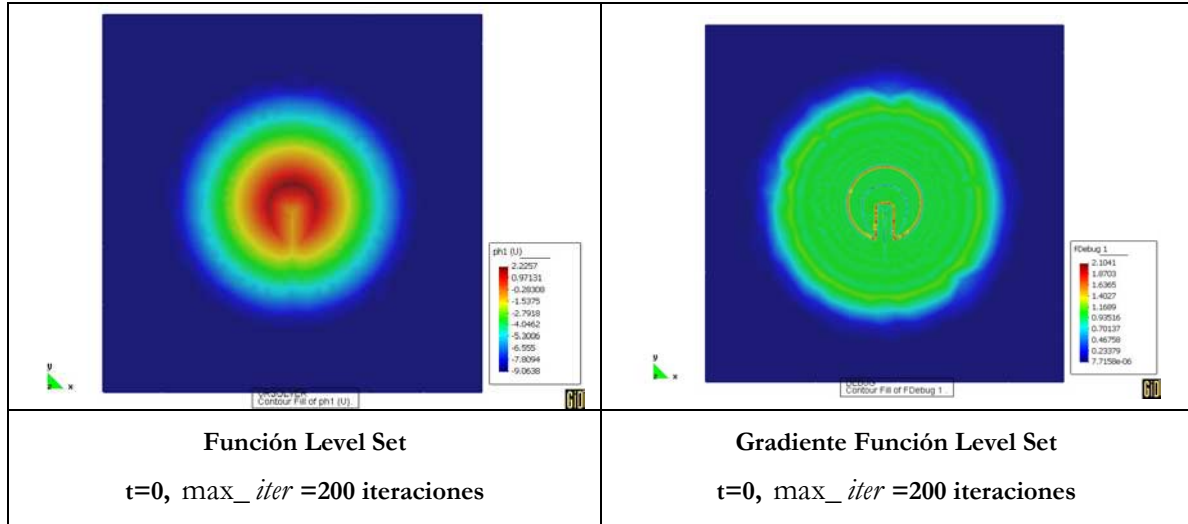
En particular se puede observar como las intersecciones de la función no coinciden en los dos gráficos, y por consiguiente la posición de interfaz, después de una vuelta, es incorrecta.



Respecto a la conservación de volumen, se observa claramente como la integral de la función Level Set no se conserva en los dos pasos de tiempo.

Un ultimo aspecto a considera en este ejemplo es comprobar que efectivamente la función Level Set obtenida por el proceso de reinicialización tiene gradiente unitario y que efectivamente la función es reinicializada en primer lugar cerca de la interfaz como se comento en la sección 2.1.1.





En la secuencia de imágenes anteriores se puede comprobar como la función efectivamente tiene gradiente unitario aproximadamente (imágenes columna derecha) cerca de interfaz y que a medida que aumentamos el número de iteraciones en el algoritmo de reinicialización la banda en que la función Level Set es inicializada con gradiente unitario es cada vez más ancha. Evidentemente en todos estos ejemplos se ha utilizado el mismo paso de pseudotiempo,  $\Delta\tau = 0.5 \cdot l_{\min(\Omega)}$ .

Es importante hacer notar que fijar un número de iteraciones muy alto aumenta considerablemente el coste computacional asociado al método Level Set respecto al tiempo total de la simulación. Es importante fijar un valor de iteraciones que busque un compromiso entre el coste computacional y la calidad de la función Level Set obtenida en la reinicialización. Como se observó la sección 2.1.1 es solo necesario tener la función Level Set definida como una función distancia con signo cerca de la interfaz. Adicionalmente se ha observado que un número de iteraciones muy elevado puede producir que los errores debidos a la difusividad numérica se vean amplificados.

Por último notar que el hecho de reinicializar la función Level Set cerca de la interfase en primer lugar permite poder aplicar esta metodología a una banda cercana a la interfase como se presenta en la sección 3.

## 8.2 Transporte burbuja

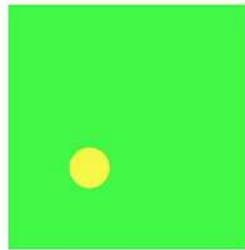


figura 16

En este ejemplo se considera el transporte de una burbuja de fluido en un dominio cuadrado lleno con el mismo tipo de fluido. La geometría del problema se describe en figura 17. Todas las magnitudes están expresadas en Sistema Internacional de unidades (m, s).

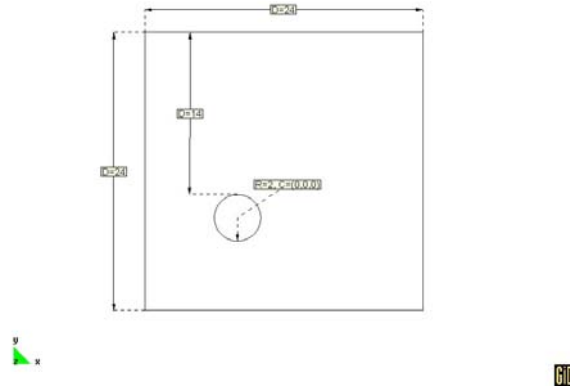


figura 17

Se ha impuesto el siguiente campo de velocidad:

$$v = \begin{cases} (2, 2) & 0 \leq t < 4 \\ (-2, -2) & 4 \leq t \leq 8 \end{cases} \quad (8.4)$$

Dado que el campo de velocidades impuesto es autónomo a trozos, no depende del tiempo, sabemos que la solución del transporte de la burbuja es equivalente a la translación de la posición inicial de interfaz.

Como en  $t=4$  seg. el sentido del vector velocidad cambia, al posición final de la interfaz debe ser la misma que la posición inicial.

Se han considerado incrementos de tiempo de 0.01 segundos, 800 incrementos para la simulación completa.

Un ejemplo parecido se presenta en [10], donde se pone de manifiesto la importancia del esquema de integración temporal. En [10] se compara dos esquemas de integración temporal par la ecuación de transporte.

- 1- Backward Euler: Esquema de integración implícito incondicionalmente estable, primer orden en el tiempo. Los resultados obtenidos con este esquema son sobredifusivos y como consecuencia no se conserva ni la forma ni el volumen inicial de interfaz.
- 2- Crank-Nicholson: Esquema de integración implícito incondicionalmente estable, primer segundo en el tiempo. Los resultados obtenidos con este esquema son correctos y se conserva la forma y el volumen inicial de interfaz.

En [10] se justifica este fenomeno debido a la mayor difusividad numérica asociada al método Backward Euler.

Nos plantemos ahora si con el uso del método de Level Set esta “artificial” dependencia con el esquema de integración temporal aún existe.

A continuación se detallan los datos de la simulación utilizando el método Level Set.

Los parámetros de los distintos algoritmos descritos en la sección 7 se han ajustado con los siguientes valores:

- Frecuencia para la Reinicialización.  $t_{init}$  : 100 iteraciones (y antes del paso de tiempo inicial).
- Paso de pseudotiempo,  $\Delta\tau$  :  $0.5 \cdot l_{\min(\Omega)}$
- Numero de pasos de pseudotiempo,  $\max\_iter$  : 20
- Parámetro Función Heaviside,  $\varepsilon$  :  $1.5 \cdot l_{\min(\Omega)}$
- Iteraciones Máximas Conservación de Volumen,  $Vol\_iter$  : 5
- Tolerancia Conservación de Volumen,  $Vol\_tol$  :  $1e-5$

La malla generada ha sido de tipo no estructurado (véase figura 19). Sean han asignado tamaños especiales a las líneas que define la posición inicial de la interfaz (0.2m), para el resto

de dominio computacional se ha utilizado un tamaño de elemento general de 0.3 (véase figura 18)

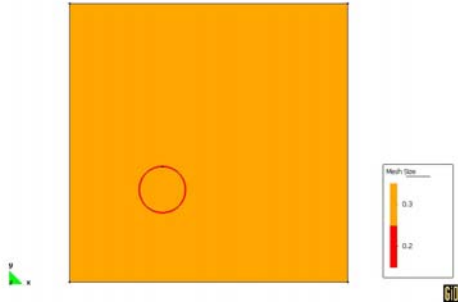


figura 18

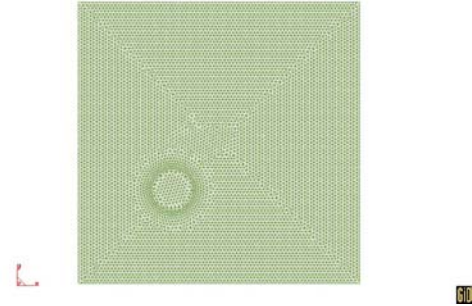
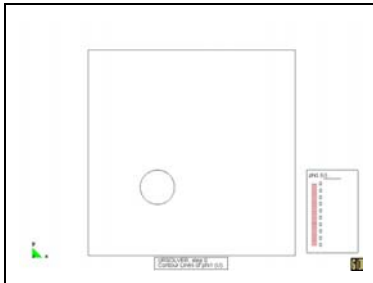
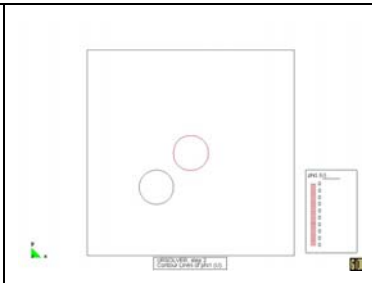
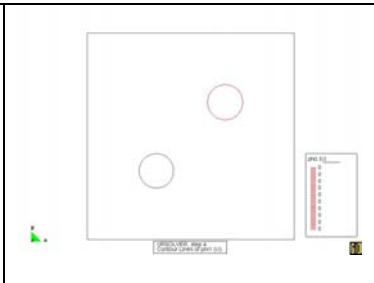
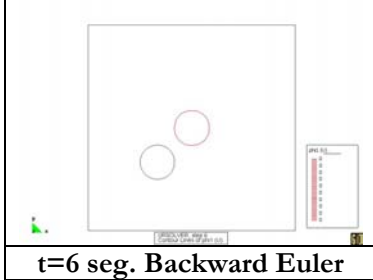
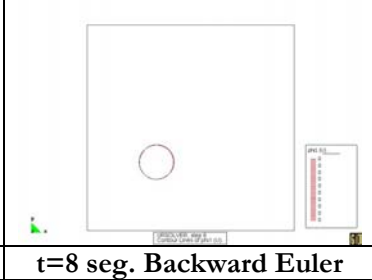


figura 19

El objetivo de malla con un tamaño menor las líneas que definen la posición inicial de la interfaz es poder aproximarla de forma mas exacta.

		
<b>t=0 seg. Backward Euler</b>	<b>t=2 seg. Backward Euler</b>	<b>t=4 seg. Backward Euler</b>
		
<b>t=6 seg. Backward Euler</b>	<b>t=8 seg. Backward Euler</b>	

La secuencia de imágenes anteriores corresponde a la evolución del frente en los instantes de tiempo 0, 2, 4, 6, 8 seg. respectivamente. Para esta simulación se utilizó Backward Euler como esquema de integración temporal. Como se puede observar la forma y el volumen de la interfaz se ha conservado perfectamente durante toda la evolución temporal. No se ha observado ninguno de los problemas comentados en [10]. Como se puede apreciar en figura 20 con más detalle la forma final ( $t=8$  seg.) de la interfaz (en rojo) frente a la forma teórica final (en negro) son casi coincidentes.

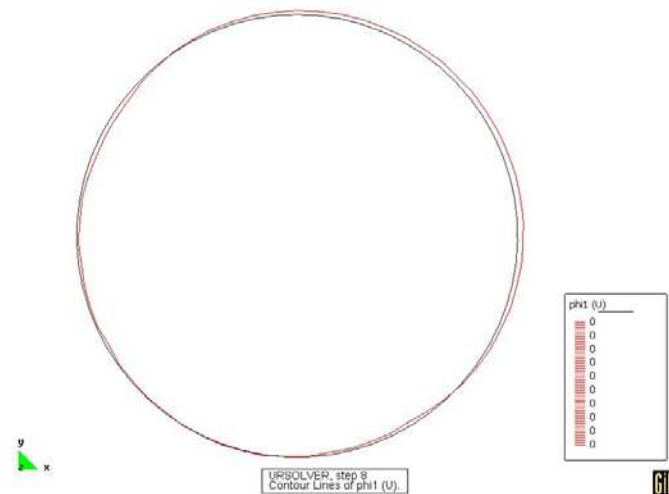
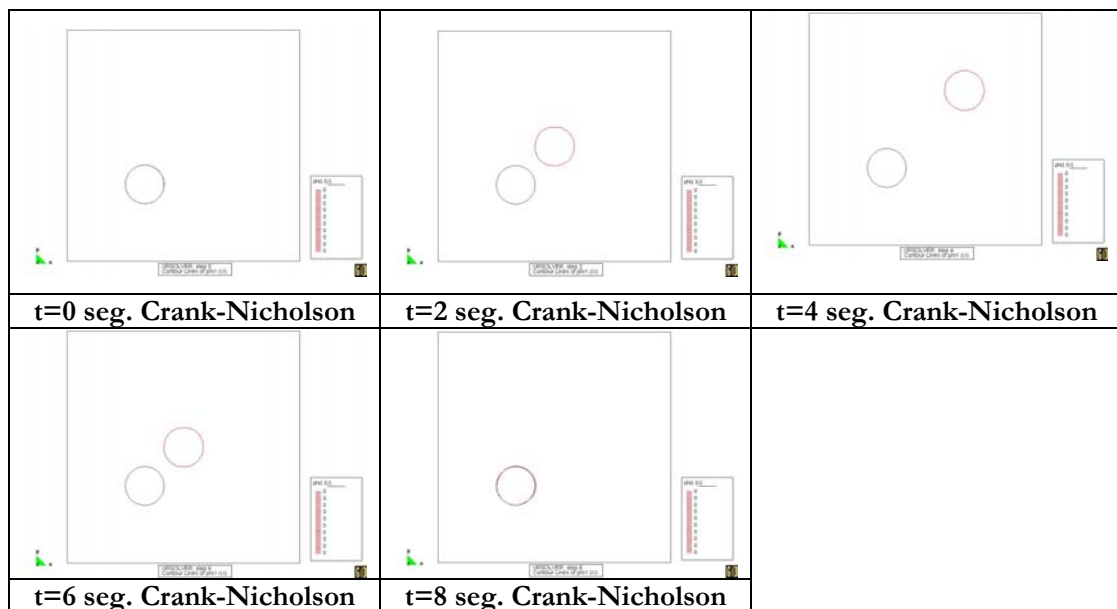


figura 20

La secuencia de imágenes siguiente es análoga a la anterior pero utilizando Crank-Nicholson como esquema de integración temporal.



Como se puede observar los resultados son casi idénticos a los obtenidos en el caso anterior utilizando Backward Euler. La posición final de la interfaz también es casi igual que a la posición final ( $t=8$  seg.) con Backward Euler como se puede ver en figura 21. Las diferencias en ambos casos son tan mínimas que se pueden considerar despreciables. Se observa que la conservación de volumen es mas precisa en el caso de Crank-Nicholson que el Backward Euler.

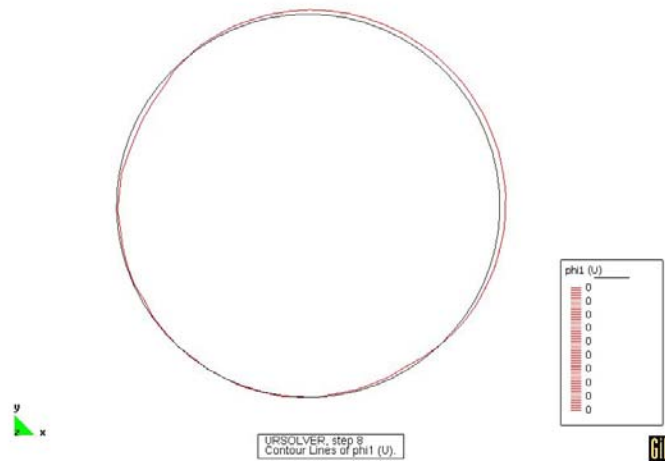


figura 21

En conclusión podemos afirmar que el uso del método Level Set permite eliminar toda independencia respecto al esquema de integración utilizado para la ecuación de transporte.

Así podemos asegurar conservación de forma y volumen independientemente del esquema de integración temporal.

Esta propiedad postulada, a posteriori, a la vista de los resultados anteriores sobre el método Level Set permite romper con la dependencia “artificial” entre forma y volumen con el esquema de integración temporal postulada en [10]. A la vez esta propiedad es de utilidad para poder descartar posibles fuentes de errores numéricas durante la evolución temporal de frente.

### 8.3 Elevación burbuja de gas en líquido

En este ejemplo se considera la elevación de una burbuja de gas sumergida en líquido. Experimentos parecidos se presentan en [18, 19, 20]. La simulación se ha realizado durante 10 seg. con incrementos de tiempo de 0.01 seg.

Las propiedades del líquido y el gas se relacionan como sigue:

$$\begin{aligned} \frac{\rho_{\text{liquido}}}{\rho_{\text{gas}}} &= 1000/1 \\ \frac{\nu_{\text{liquido}}}{\nu_{\text{gas}}} &= 100/1 \end{aligned} \tag{8.5}$$

La posición inicial y geometría del problema se presentan en figura 22. Todas las magnitudes están expresadas en Sistema Internacional de unidades (m,s,Kg)

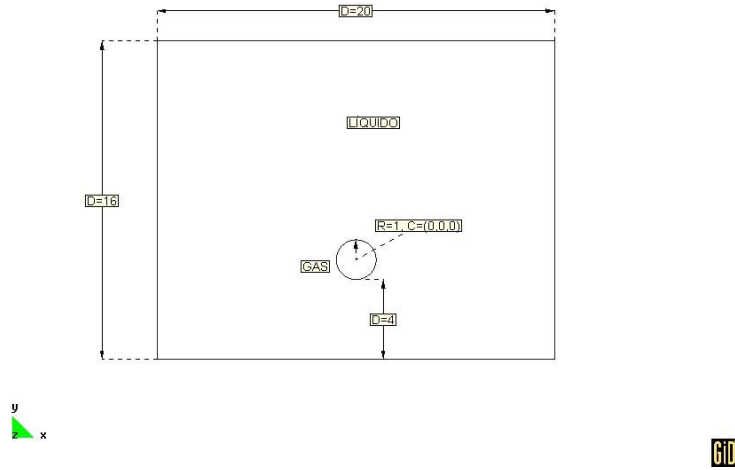


figura 22

Todos los contornos se han definido como contornos impenetrables (con velocidad normal nula impuesta).

Se espera que dada la diferencia de densidades entre el gas y el líquido, la burbuja de gas se eleve.

Los parámetros de los distintos algoritmos descritos en la sección 7 se han ajustado con los siguientes valores:

- Frecuencia para la Reinicialización,  $t_{init}$  : 100 iteraciones (y antes del paso de tiempo inicial).
- Paso de pseudotiempo,  $\Delta\tau$  :  $0.5 \cdot l_{\min(\Omega)}$
- Numero de pasos de pseudotiempo,  $\max\_iter$  : 20
- Parámetro Función Heaviside,  $\varepsilon$  :  $1.5 \cdot l_{\min(\Omega)}$
- Iteraciones Máximas Conservación de Volumen,  $Vol\_iter$  : 5
- Tolerancia Conservación de Volumen,  $Vol\_tol$  :  $1e-5$

La malla generada ha sido de tipo no estructurado (véase figura 24). Se han asignado tamaños especiales a las líneas que define la posición inicial de la interfaz (0.1m), para el resto de dominio computacional se ha utilizado un tamaño de elemento general de 0.3 (véase figura 23)

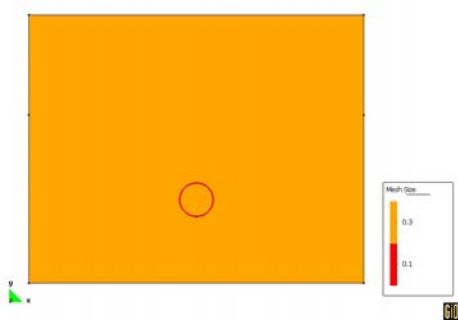


figura 23

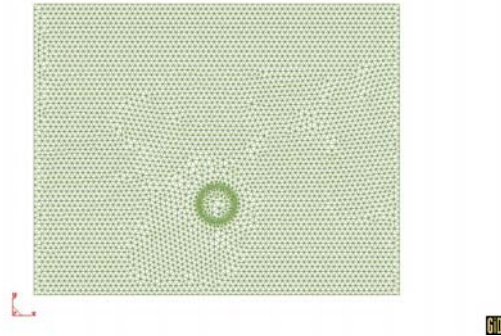


figura 24

El número de nodos es de 4377 y el total de elementos generados es de 8815.

En tabla 3 se presenta una secuencia de imágenes de la evolución de la burbuja en el líquido. Se puede apreciar como en el paso de tiempo  $t=3$  seg. la burbuja rompe en dos partes, dado

que la estabilidad de la forma de burbuja depende de la relación entre las propiedades del fluido y el gas y el tamaño y forma original de la burbuja.

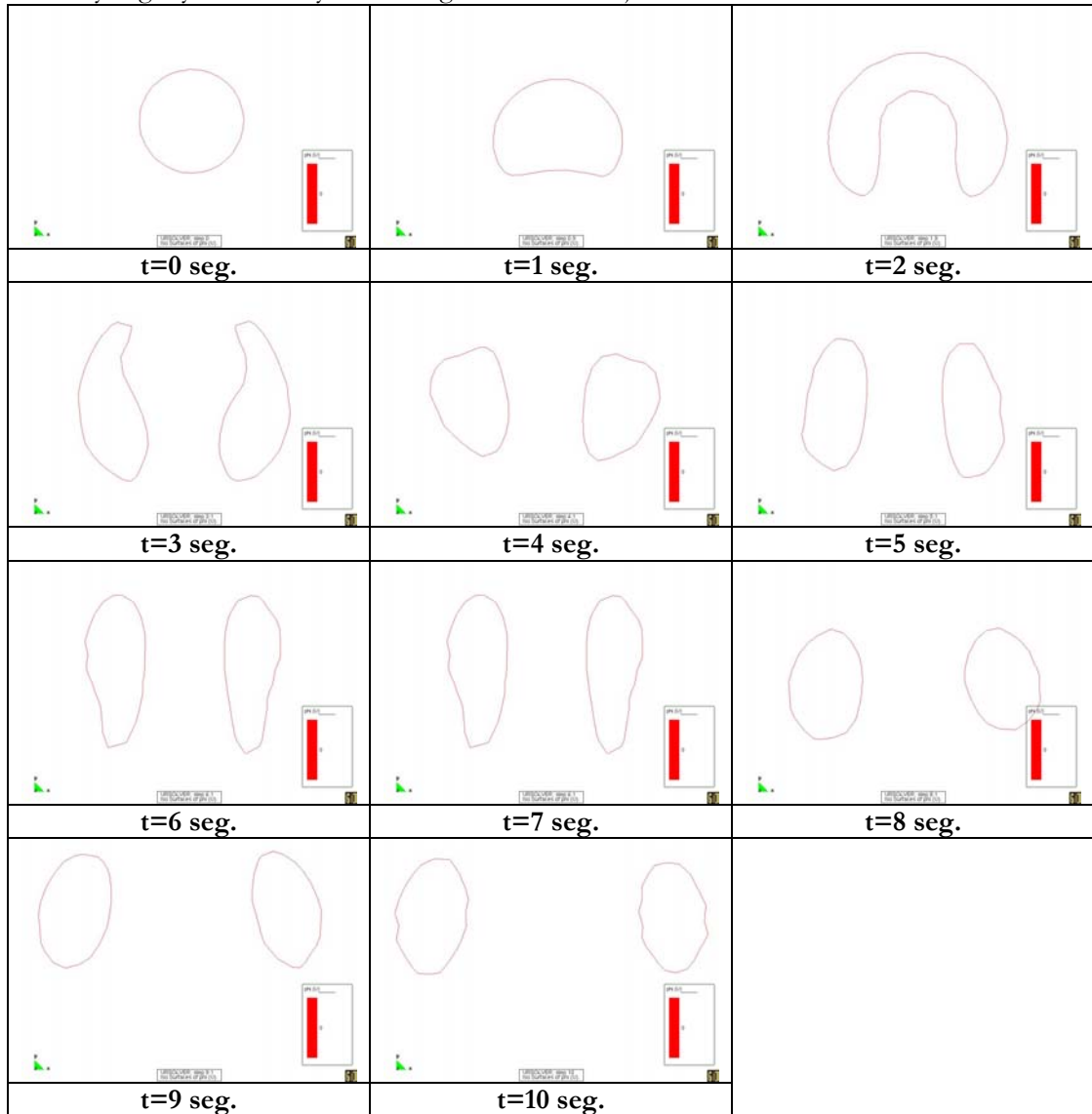


tabla 3

Los resultados obtenidos son muy similares a los presentados por Sussman en [18] con la ventaja adicional del uso de un malla no estructurada con un número de nodos mucho menor. En los experimentos de Sussman se utilizan mallas estructuradas de triángulos de 100x200 nodos por lados, unos 20.000 nodos.

#### 8.4 Sector circular

En este ejemplo se estudia la influencia de la aproximación inicial de interfaz sobre el método Level Set.

En todos los ejemplos anteriores la posición inicial de la interfaz estaba totalmente determinada y la malla se generaba con elementos más pequeños cerca de interfaz y con nodos situados encima de la interfaz (con valor cero de la función Level Set). Esta situación no tiene porque ser general y pueden encontrarse casos en que generar una malla con estas características no sea posible.

En este ejemplo se generan dos mallas:



- Una malla no estructurada de triángulos, sin colocar nodos encima de la interfaz ni reducir el tamaño de los elementos cerca de interfaz con el objetivo de mejorar la aproximación inicial de la interfaz.
- Una malla no estructurada de triángulos, pero esta vez con nodos situados en el frente y aumentado el número de elementos cerca de la interfaz.

El objetivo es comprobar la influencia de la calidad de aproximación inicial sobre el método. El problema es parecido al presentado en el ejemplo 8.1, la rotación de sólido rígido de un sector circular. Todos los datos del problema son los mismos usados en el ejemplo 8.1. En la figura 25 se describe la geometría del problema. Todas las magnitudes están expresadas en Sistema Internacional de unidades (m, s).

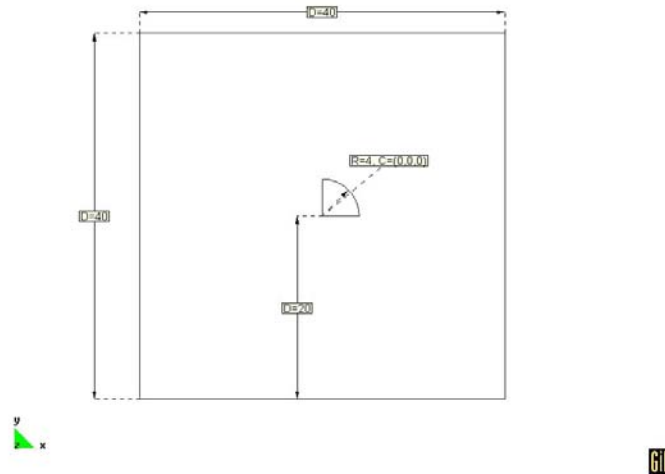


figura 25

Las características de las mallas generadas son:

- Malla no ajustada al frente: Tamaño general de elemento de 0.2 m es todo el dominio.
  - Nodos: 46444
  - Elementos Triangulares: 92086
  - Elementos Lineales: 800
- Malla no ajustada al frente: Tamaño de elemento de 0.1 m para las líneas que definen la posición inicial del frente. Tamaño general de elemento de 0.2 m para el resto del dominio (véase figura 26).
  - Nodos: 49593
  - Elementos Triangulares: 92384
  - Elementos Lineales: 943

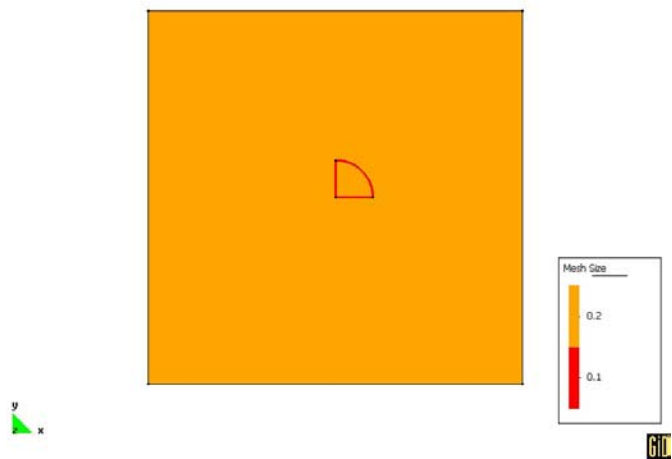


figura 26

	Malla No Ajustada	Malla Ajustada
t=0 seg.		
t=4 seg.		

tabla 4

Como se observa en tabla 4 la aproximación inicial para la malla no ajustada presenta oscilaciones en todo el contorno, así como un redondeo inicial de las esquinas de la interfaz. En cambio la aproximación inicial para la malla ajustada recoge sin oscilaciones los lados rectos de la interfaz, respeta mucho mas la forma de las esquinas y las oscilaciones presentes en el arco de circunferencia son menores. En figura 27 se muestra con detalle la aproximación de ambas mallas de la posición inicial de la interfaz.

La forma del frente después de una vuelta (t=4 seg.) presenta diferencias importantes entre las dos mallas. Para la malla no ajustada el redondeo inicial de la esquinas se ha visto amplificado de forma muy significativa y la oscilaciones iniciales en la interfaz aun persisten

en algunas zonas. En el caso de la malla ajustada, aún y presentar un redondeo importante de la forma original de la interfaz, la forma original de la interfaz se conserva mejor. Se observa también que el contorno es mas “suave” o regular presentando muy pocas oscilaciones. Es importante destacar que donde se han localizado mas errores en la conservación del contorno inicial de la interfaz es donde inicialmente se presentaban mayores oscilaciones y por consiguiente mayores errores de aproximación.

Un comentario para ambos casos y como ya se observo en el ejemplo 8.1 el método Level Set tiene tendencia a resultar sobredifusivo. Se ha comprobado durante la realización de estos ejemplos que es de vital importancia ajustar el número de veces que es aplicado el algoritmo de reinicialización para llegar a un compromiso entre la conservación de la propiedad de función distancia con signo y la difusividad numérica asociada este proceso.

A la vista de los resultados obtenidos, se puede concluir que siempre que sea posible disponer de información sobre la posición inicial de la interfaz será muy recomendable generar una malla adaptada a esta posición.

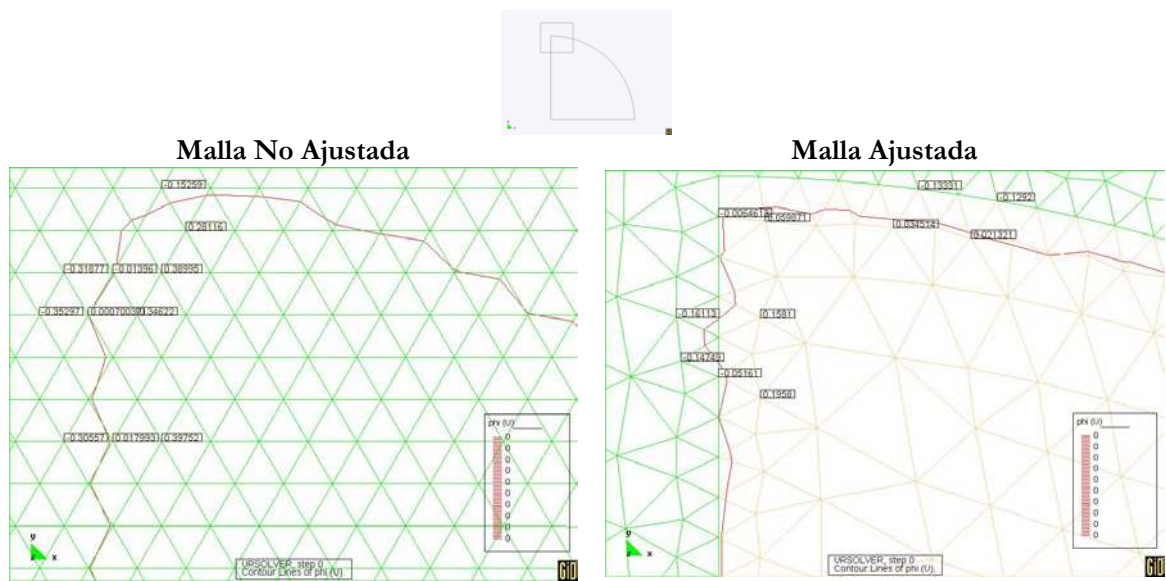


figura 27

## 8.5 Rotación sólido rígido esfera Zalesak

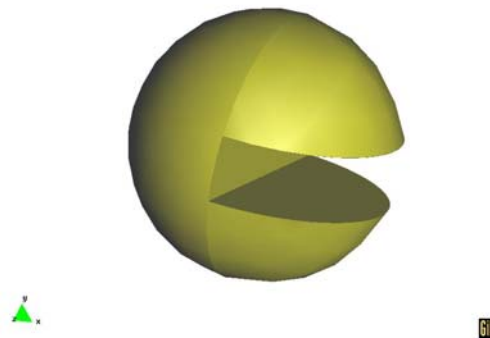


figura 28

En este ejemplo se considera la rotación de sólido rígido de una esfera con una hendidura, como en la figura 28. Este ejemplo es la generalización al caso tridimensional del ejemplo 8.1

La geometría de este ejemplo se describe en figura 12, todas las magnitudes están expresadas en Sistema Internacional de unidades (m,s).

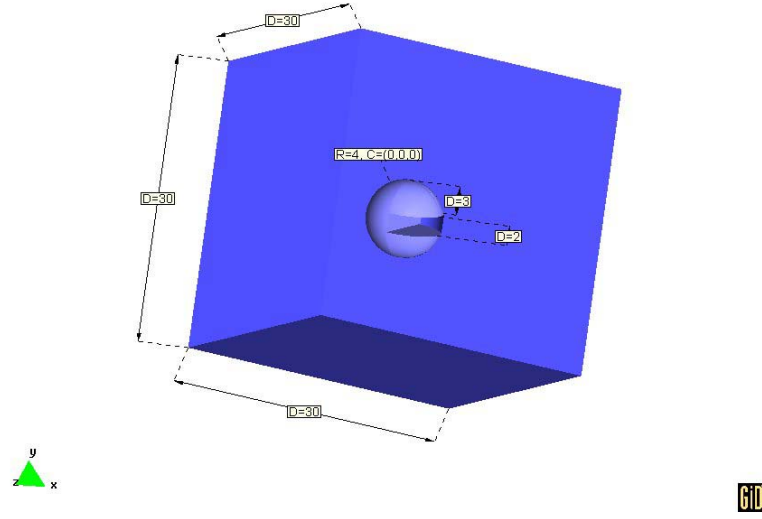


figura 29

El campo de vorticidad constante impuesto en el dominio computacional es:

$$\begin{aligned} v_x &= \frac{\pi}{2} y \\ v_y &= -\frac{\pi}{2} x \\ v_z &= 0 \end{aligned} \quad (8.6)$$

Este campo de vorticidad define una rotación de eje OZ con velocidad angular  $\frac{\pi}{2}$ .

Entonces para realizar una rotación completa  $2\pi \text{ rad}$  se necesitaran 4 segundos exactamente. Se han considerado incrementos de tiempo de 0.01 segundos, 400 incrementos para una rotación completa.

Como el campo de velocidades impuesto es estacionario, no depende del tiempo, sabemos que la solución de la ecuación de transporte cumple:

$$\phi(x, t) = \phi_0(x - ut) \quad (8.7)$$

Donde  $\phi_0(x)$  es la posición inicial de la interfaz. Así como el campo de velocidades impuesto es una rotación en sentido contra horario de eje OZ, es de esperar que la evolución de la interfaz sea equivalente a la rotación de sólido rígido de la posición inicial de la interfaz, o sea de la esfera entorno al eje OZ.

Para la simulación se ha utilizado una malla de tetraedros de 3 nodos. La malla generada ha sido de tipo no estructurado y se han asignado tamaños especiales a las superficies que define la posición inicial de la interfaz (0.1 m) así como a los volúmenes interiores a la esfera (0.6 m) (véase figura 30), para el resto de dominio computacional se ha utilizado un tamaño de elemento general de 2 m. En relación al caso bidimensional, ejemplo 8.1, la malla generada para este ejemplo tridimensional pretende ser una generalización de la malla 2-D. Se han asignando los mismos tamaños de elemento a las entidades que generalizan el caso bidimensional, líneas por superficies y superficies por volúmenes.

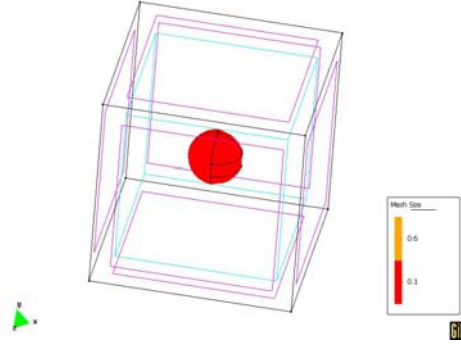
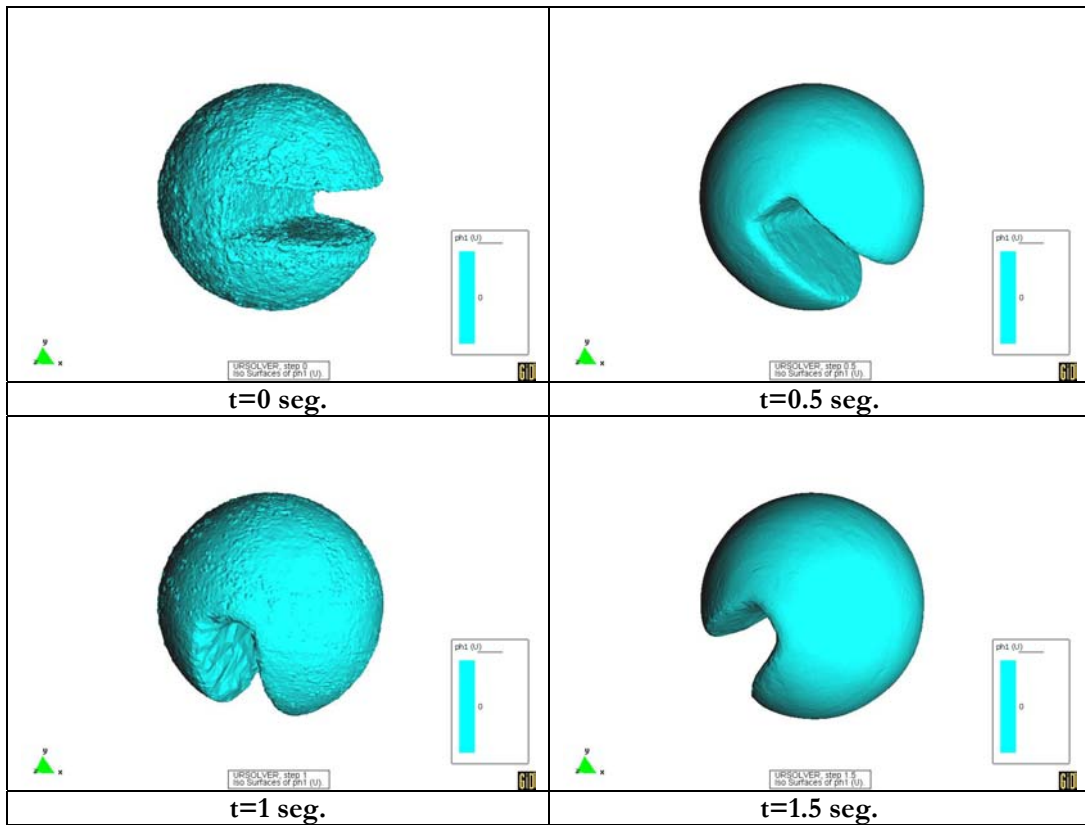
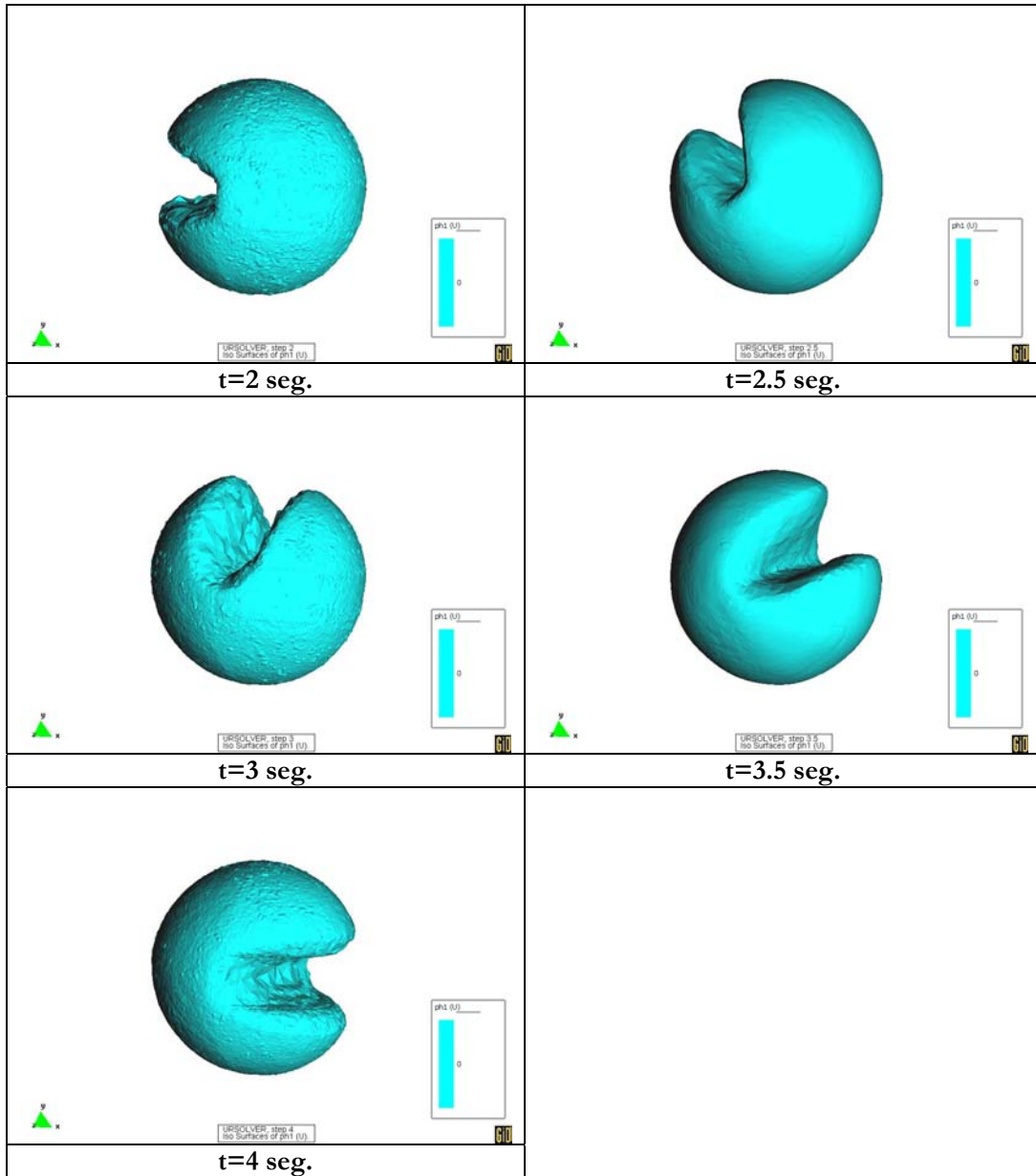


figura 30

Los parámetros de los distintos algoritmos descritos en la sección 7 se han ajustado con los siguientes valores:

- Frecuencia para la Reinicialización.  $t\_init$  : 100 iteraciones (y antes del paso de tiempo inicial).
- Paso de pseudotiempo,  $\Delta\tau$  :  $0.5 \cdot l_{\min(\Omega)}$
- Numero de pasos de pseudotiempo,  $max\_iter$  : 15
- Parámetro Función Heaviside,  $\varepsilon$  :  $1.5 \cdot l_{\min(\Omega)}$
- Iteraciones Máximas Conservación de Volumen,  $Vol\_iter$  : 5
- Tolerancia Conservación de Volumen,  $Vol\_tol$  :  $1e-5$





En la secuencia de imágenes anterior se muestra la evolución de la posición de la esfera a lo largo de una rotación completa.

Se observa que el método Level Set ofrece también muy buenos resultados en problemas tridimensionales. Como se adelantó en secciones anteriores, una muy buena propiedad de este método es que su generalización al caso tridimensional es trivial, únicamente requiere desarrollar algoritmos que acepten elementos tridimensionales.

Vemos que el algoritmo de conservación de volumen también funciona de forma correcta en este caso de dimensión superior.

Dado que el proceso de reinicialización se realiza cada 100 pasos de tiempo y cada paso de tiempos es 0.01 seg., este proceso se aplica en  $t=0, 1, 2, 4$  seg. (Columna izquierda). Como se puede ver en las imágenes de la columna de la izquierda, el proceso de reinicialización produce que aparezcan pequeñas irregularidades (rugosidades) en la superficie de la interfaz. Este hecho se debe a que la función Level Set obtenida de la reinicialización depende de la calidad de la malla, el tamaño de las irregularidades tiene orden el tamaño de elemento adyacente a la interfaz. (Véase figura 31). Así es posible minimizar estas rugosidades generando una malla con elementos pequeños en las zonas adyacentes al interfaz.

De todos modos estas irregularidades no son de gran importancia, dado que en los siguientes pasos de tiempo son eliminadas por efecto de la difusividad del propio método, como se puede ver en los pasos de tiempo intermedios  $t=0.5, 1.5, 2.5, 3.5$  seg., mostrados en la secuencia de imágenes anterior, donde la superficie de la interfaz aparece totalmente lisa.

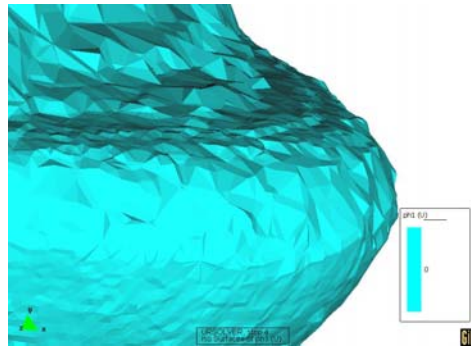


figura 31

En tabla 5 se presenta los resultados de la restricción de la interfaz al plano XY. Este problema es equivalente al ejemplo 8.1, dado que la restricción de la interfaz al plano XY reduce la dimensión del problema. La interfaz es ahora un disco con una hendidura.

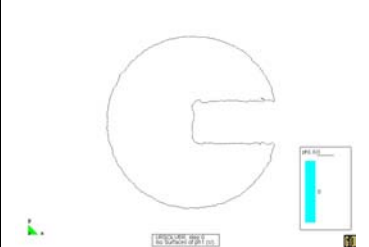
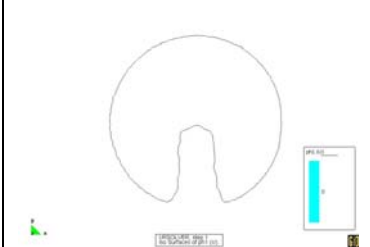



		
<b>t=0 seg.</b>	<b>t=1 seg.</b>	<b>t=2 seg.</b>
		
<b>t=3 seg.</b>	<b>t=4 seg.</b>	

tabla 5

Los resultados obtenidos no son tan precisos como en el caso bidimensional, pero la topología del frente se conserva a diferencia de cuando no se usa el método Level. Se puede concluir que los resultados obtenidos para este caso tridimensional son de alta calidad y que la generalización del método Level Set al caso tridimensional es trivial.

## 8.6 Superficie Libre

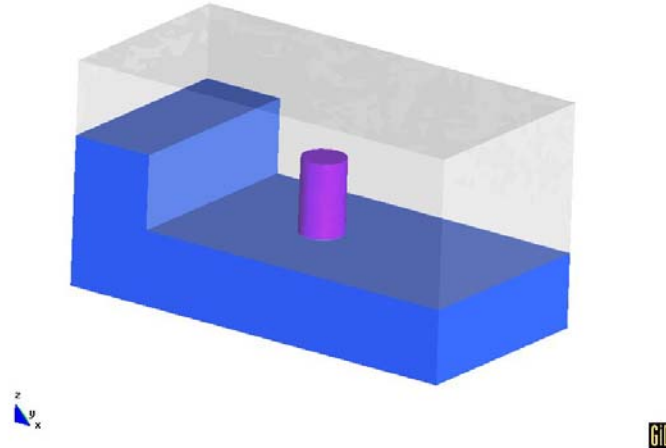


figura 32

En este ejemplo se presentan los resultados de la simulación numérica de la interfaz entre dos fluidos contenidos en un domino prismático tridimensional con un cilindro sólido parcialmente sumergido en su interior. Es habitual denominar por superficie libre la interfaz entre fluidos en el ámbito de la mecánica de fluidos y la ingeniería naval.

Aunque los resultados son difícilmente contrastables dado que no se disponen de datos experimentales, este ejemplo es de utilidad ya que permite mostrar las buenas propiedades de este método para la solución de problemas reales en el campo de la ingeniería y la mecánica de fluidos.

Dado la naturaleza del problema planteado es necesaria la solución de las ecuaciones de Navier-Stokes para obtener el campo de velocidades necesario para la evolución de interfaz. Este hecho pondrá de manifiesto el problema asociado a la solución de las ecuaciones de Navier-Stokes cuando la densidad y la viscosidad son discontinuas en el dominio computacional. Veamos cual es el origen del problema.

Supongamos que se quiere resolver el siguiente problema simplificado:

Un flujo hidrostático en un domino cuadrado bidimensional,  $\Omega$ , con dos fluidos en equilibrio.

Como el problema es hidrostático, las ecuaciones Navier-Stokes se reducen a

$$\nabla p = \rho g \quad (8.8)$$

Donde  $g$  es la gravedad.

Dado que tengo dos fluidos, la densidad del flujo en el dominio  $\Omega$  es discontinua

$$\rho_{\text{flujo}}(x) = \begin{cases} \rho_1 & x \in \Omega_1 \\ \rho_2 & x \in \Omega_2 \end{cases} \quad (8.9)$$

Donde  $\Omega_i = \{x \in \Omega \mid x \text{ es fluido-}i\}$   $i = 1, 2$  y  $\rho_i$  es la densidad asociada al fluido  $i$ .

Así se cumple que

$$\nabla p(x) = \begin{cases} \rho_1 g & x \in \Omega_1 \\ \rho_2 g & x \in \Omega_2 \end{cases} \quad (8.10)$$

Se deduce de (8.10) que el gradiente del campo de presiones es discontinuo.

A nivel discreto, para poder reproducir de forma exacta el campo de presiones, es necesario poder aproximar un gradiente de presión discontinuo. La única forma de poder reproducir un gradiente discontinuo de presión es generar una malla en que ningún elemento este



cortado por la interfaz, es decir, una malla en que los elementos adyacentes a la interfaz tengan los nodos exactamente sobre la interfaz. Vemos porque:

Para elementos finitos “clásicos” bidimensionales, las funciones de forma elementales son polinomios en dos variables de distintos grados, en función del orden del elemento. En particular el gradiente de polinomios es una función continua.

Así se tiene, para un elemento  $e$  fijo

$$\begin{aligned} p &\simeq p_b = \sum_i N_i(x) p_i \\ \nabla p &\simeq \nabla p_b = \sum_i \nabla N_i(x) p_i \end{aligned} \quad (8.11)$$

Donde  $N_i$  son las funciones de forma elementales para interpolar la presión.

Se deduce de (8.11) que el gradiente de presión se aproxima por una combinación lineal de funciones continuas,  $\nabla N_i$ , en particular, será continuo en el elemento  $e$ .

Por tanto, con los elementos finitos clásicos es imposible reproducir un gradiente de presión discontinuo como en (8.10).

La única forma de poder reproducir un gradiente de presión discontinuo en  $\Omega$ , utilizando elementos finitos clásicos, es imponer que los nodos de los elementos adyacentes a la interfaz estén descansando encima la interfaz, dado que la interpolación del gradiente de presión entre elementos si puede ser discontinua.

Volviendo a la presentación del ejemplo, en figura 33 se describe la geometría del problema, todas las unidades están en Sistema Internacional (m, kg, s).

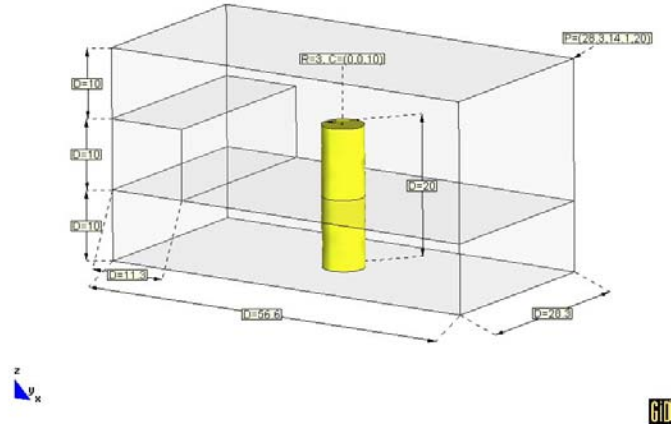


figura 33

La simulación se ha realizado durante 20 seg. con incrementos de tiempo de 0.01 seg. Las propiedades de los fluidos se relacionan como sigue:

$$\begin{aligned} \rho_1 / \rho_2 &= 1000 / 1 \\ \nu_1 / \nu_2 &= 100 / 1 \end{aligned} \quad (8.12)$$

Donde  $\nu_i$  es la viscosidad del fluido  $i$

La posición inicial de la interfaz se describe como (véase figura 32):

$$\phi_0 = \begin{cases} 1 & x \leq 11.3 \quad \tilde{z} \leq 10 \\ 1 & x > 11.3 \quad \tilde{z} \leq 0 \\ -1 & \text{en caso contrario} \end{cases} \quad (8.13)$$

Para poder solucionar el problema con la presión se ha decidido resolver únicamente uno de los fluidos (el de mayor densidad y viscosidad,  $\phi \geq 0$ ) en todo dominio computacional. Con este objetivo se ha definido un campo gravitatorio como sigue:

$$g(x, t) = \begin{cases} (0, 0, -9.8) & \phi(x, t) \geq 0 \\ (0, 0, 0) & \phi(x, t) < 0 \end{cases} \quad (8.14)$$

Las condiciones de contorno aplicadas para las ecuaciones Navier-Stokes son contornos impenetrables (componente normal de la velocidad nula) en todo el contorno excepto en el contorno superior del prisma, que se ha dejado libre.

Las velocidades iniciales de ambos fluidos son nulas y no hay admisión ni pérdida de flujo en ninguna parte del dominio.

Para la simulación se ha utilizado una malla de tetraedros de 3 nodos. La malla generada ha sido de tipo no estructurado y se han asignado tamaños especiales a las superficies que define la posición inicial de la interfaz (0.5 m) (véase figura 34), para el resto de dominio computacional se ha utilizado un tamaño de elemento general de 2 m.

Las características de la malla generada son:

- Nodos: 28393
- Elementos Tetraedros: 150588
- Elementos Triangulares: 28013

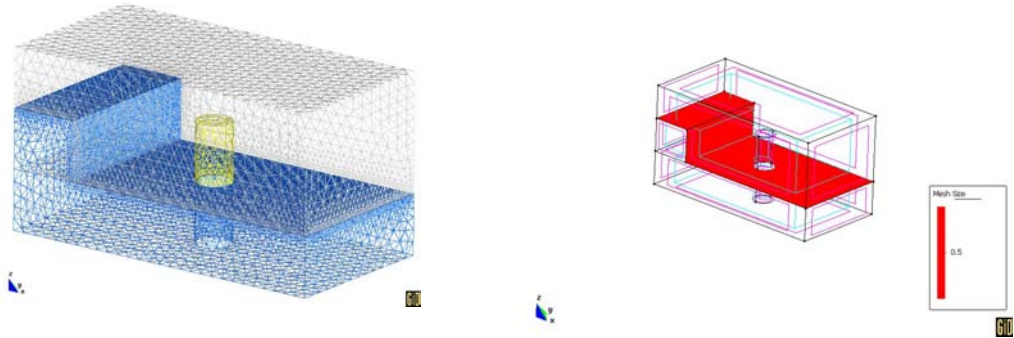


figura 34

Los parámetros de los distintos algoritmos descritos en la sección 7 se han ajustado con los siguientes valores:

- Frecuencia para la Reinicialización.  $t_{init}$ : 100 iteraciones (y antes del paso de tiempo inicial).
- Paso de pseudotiempo,  $\Delta\tau$ :  $0.5 \cdot l_{\min(\Omega)}$
- Numero de pasos de pseudotiempo,  $\max\_iter$ : 15
- Parámetro Función Heaviside,  $\varepsilon$ :  $1.5 \cdot l_{\min(\Omega)}$
- Iteraciones Máximas Conservación de Volumen,  $Vol\_iter$ : 5
- Tolerancia Conservación de Volumen,  $Vol\_tol$ :  $1e-5$

En tabla 6 se presentan los resultados obtenidos en la simulación de la evolución de la interfaz.

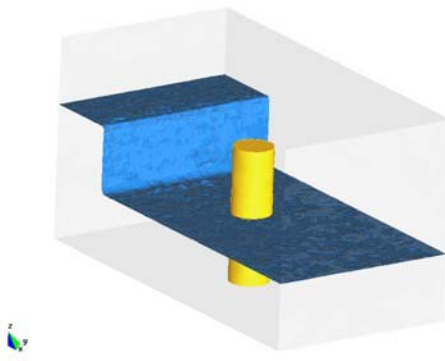
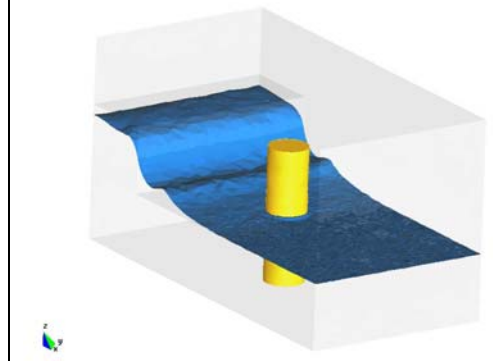
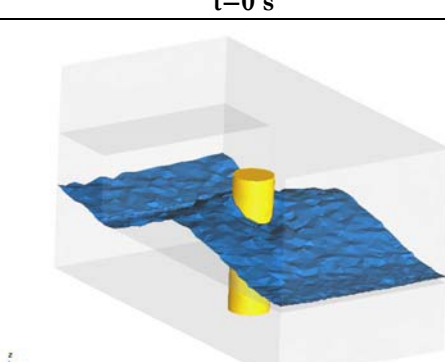
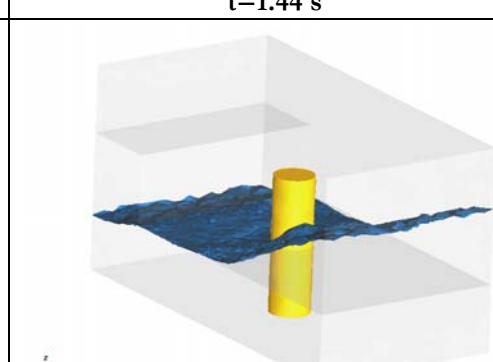
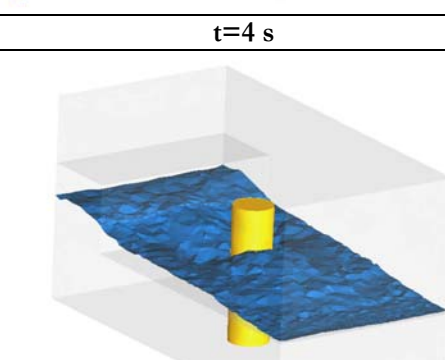
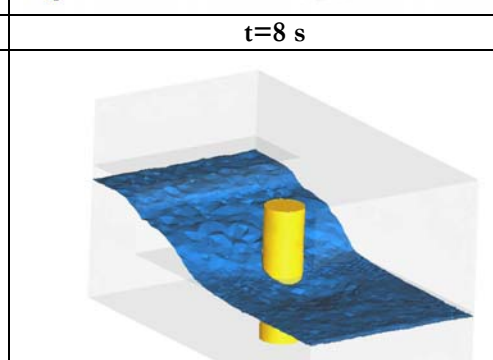
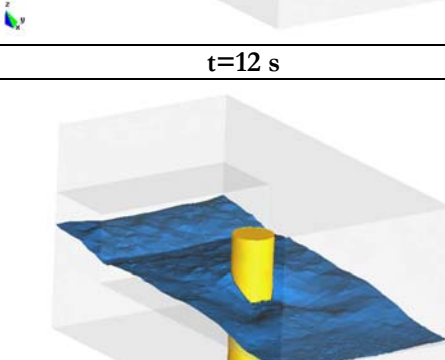
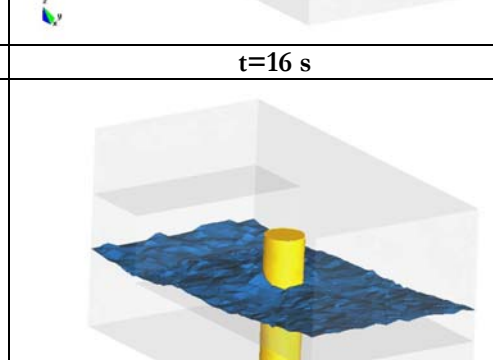
	
$t=0 \text{ s}$	$t=1.44 \text{ s}$
	
$t=4 \text{ s}$	$t=8 \text{ s}$
	
$t=12 \text{ s}$	$t=16 \text{ s}$
	
$t=18.08 \text{ s}$	$t=20 \text{ s}$

tabla 6

Dada la posición inicial del frente, la columna de fluido más denso es inestable y empieza a desplazarse por acción de la gravedad. Durante la evolución temporal se observa como se crea una ola que viaja de izquierda a derecha hasta llegar a la pared opuesta en que es reflejada en sentido opuesto.

A lo largo del tiempo la amplitud y velocidad de propagación de la ola es menor por efecto de la viscosidad. Como se puede observar los resultados obtenidos son físicamente coherentes.

A continuación en gráfico 4 y gráfico 5 se presentan la variación temporal de la función Level Set y la presión en tres nodos de la malla (véase figura 35) respectivamente:

- **nodo 13695**: Este nodo esta situado en la intersección del cilindro con el plano  $Z=0$  y el plano  $Y=0$  (coordenadas  $(-3,0,0)$ ). Así este nodo esta en la pared del cilindro.
- **nodo 8891**: Este nodo esta situado en la intersección del plano  $X=28.3$  y el plano  $Y=-14.1$  y el plano  $Z=3.374$ . Gráficamente esta en la arista vertical inferior derecha, según el plano  $Z=0$ .
- **nodo 28337**: Este nodo esta situado en la intersección del plano  $X=-28.3$  y el plano  $Y=-14.1$  y el plano  $Z=4.26$ . Gráficamente esta en la arista vertical inferior izquierda, según el plano  $Z=0$ .

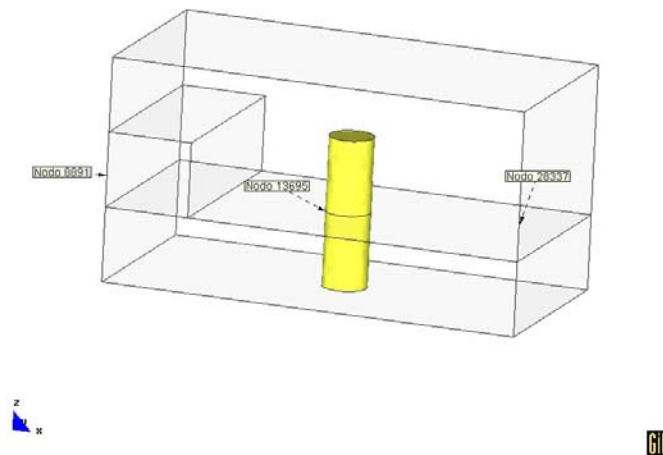


figura 35

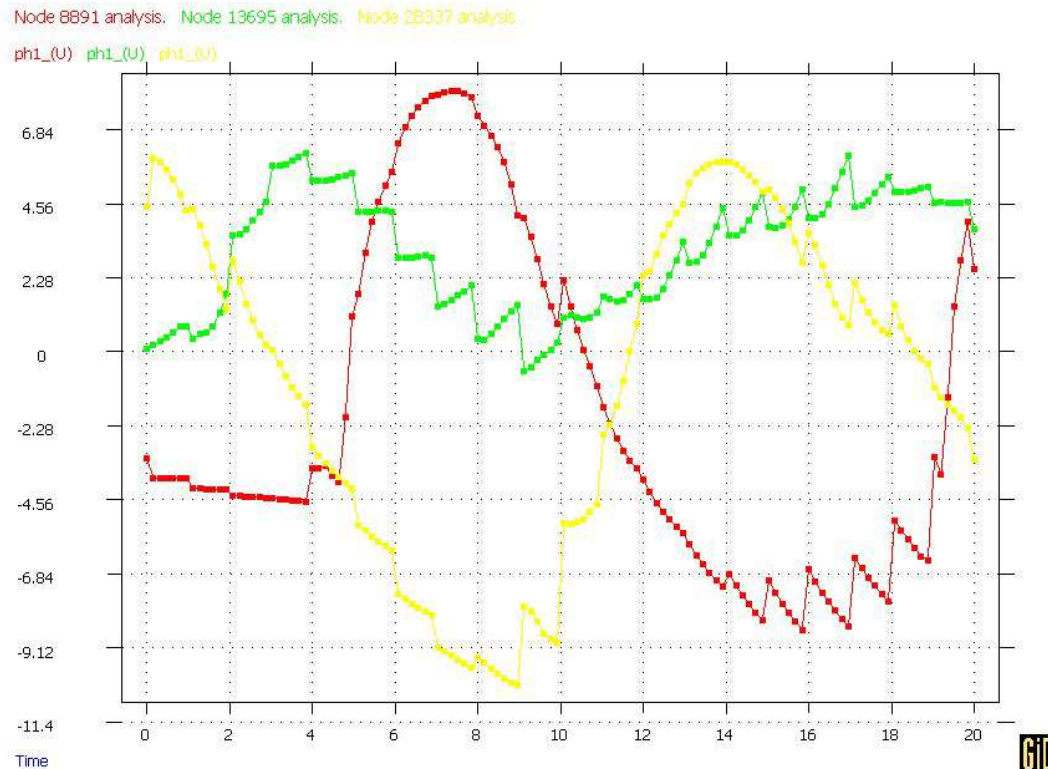


gráfico 4

Se observa como para el nodo 13695 (color verde) la evolución de la función Level Set equivalentemente la evolución de interfaz presenta importantes saltos de forma casi periódica. Durante un segundo el movimiento de la interfaz es suave y posteriormente se produce un desplazamiento relativamente muy mayor al realizado por el frente al lo largo del segundo anterior.

Este importante desplazamiento puede encontrar su explicación en el efecto de la viscosidad del fluido sobre la superficie del cilindro:

Durante un segundo se va creando una capa de fluido casi completamente en reposo pegada a la pared del cilindro, por efecto de la viscosidad del fluido. A medida que pasa el tiempo esta capa de fluido aumenta hasta llegar a su masa crítica. Entonces cuando el efecto de la viscosidad no es suficiente para mantener la capa pegada al cilindro, la capa se desprende de forma casi instantánea provocando un importante desplazamiento instantáneo de la interfaz.

Para los nodos 8891 (color verde) y 28337 (color amarillo) se observa que el comportamiento de la interfaz es bastante simétrico, lógico dadas su posiciones relativas en el dominio. Se observa también, para ambos nodos, que durante los periodos de tiempo en que la interfaz se eleva respecto su posición la evolución de la interfase es suave y continua, mientras que los periodos en que la interfaz desciende respecto su posición, el desplazamiento del frente presenta importantes saltos parecidos a los presentes en el nodo 13695. La justificación de los saltos presentes en el nodo 13695 es también válida para poder justificar estos saltos, el efecto de la viscosidad sobre las paredes del contorno.

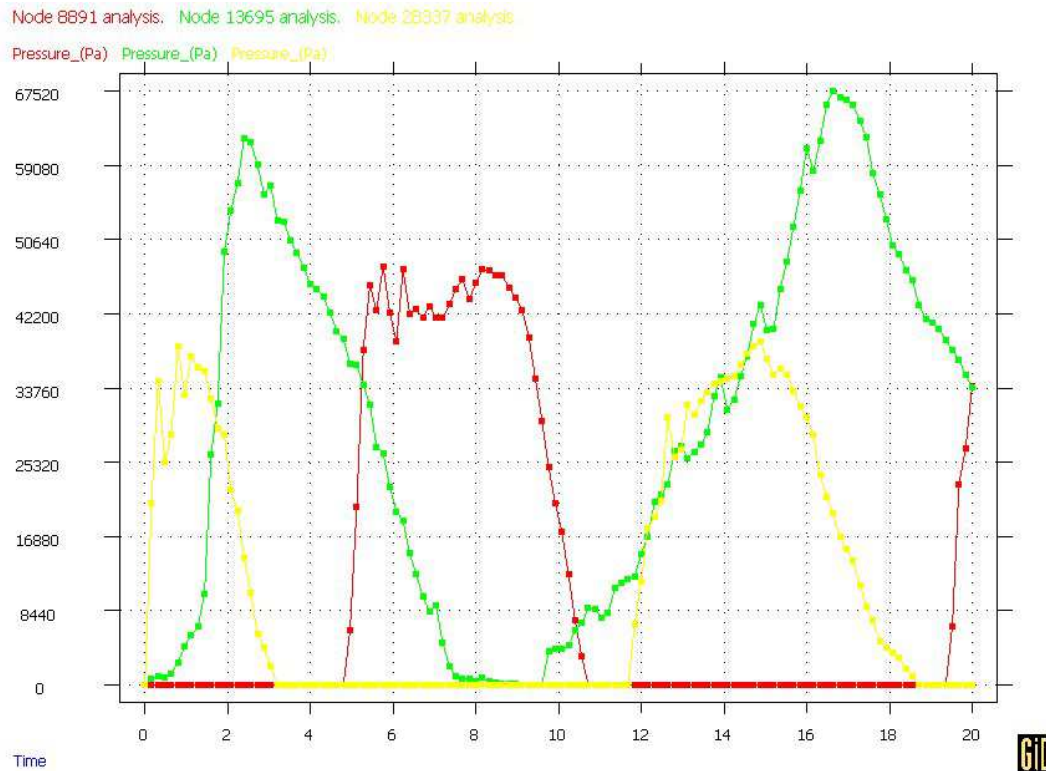


gráfico 5

En gráfico 5 se puede ver el efecto en el campo de presión debido al campo gravitatorio definido en (8.14).

Los puntos de corte de función Level Set con el eje OX del gráfico 4 son equivalentes a los instantes de cruce de la interfaz con los nodos. En estos instantes el fluido presente en el nodo cambia y consecuentemente la definición del campo gravitatorio cambia como se puede ver en gráfico 5

## 9. Conclusiones

En vista de los resultados anteriores, el método Level Set es anunciada como una herramienta de gran utilidad para el problema de evolución de interfaz entre fluidos.

Más concretamente el método Level Set mejora de forma substancial los problemas numéricos asociados al transporte de una función escalón.

Se cumple que las propiedades numéricas (regularidad) de la función Level Set, obtenida por el proceso de reinicialización, ayudan enormemente a minimizar los errores asociados al esquema de integración numérico de la ecuación de transporte (9.1).

$$\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = 0 \quad (9.1)$$

Como se observa en el ejemplo 8.1 el uso del método Level Set transfiere un grado de fiabilidad mayor a los resultados ya que respeta la topología del frente, de forma eficaz.

Durante la computación de los distintos ejemplos se ha detectado gran dependencia respecto a la frecuencia con que es aplicado el algoritmo de reinicialización, concluyéndose que una frecuencia de reinicialización muy alta acentúa los problemas de difusividad numérica presentes en la integración de (9.1) mientras que una frecuencia de reinicialización muy baja provoca la pérdida de las “buenas” propiedades numéricas de la función Level Set y consecuentemente provoca un deterioro de la forma del frente.

Así se deduce que encontrar un valor de compromiso entre estos dos fenómenos es de vital importancia para una correcta evolución del frente. De forma heurística se ha aproximado este valor en unas 4 o 6 veces a lo largo de toda la evolución.

También es importante resaltar que la dependencia de los resultados respecto al resto de parámetros no se ha mostrado notable en ningún caso, siendo aconsejable no sobredimensionar el número de pasos de pseudotiempo para no aumentar así, innecesariamente, el tiempo de cálculo.

Si merece un comentario el hecho de sobredimensionar el parámetro  $\varepsilon$  (asociado a la función Heaviside suavizada). Sobredimensionar  $\varepsilon$ , que define el ancho ficticio de la interfaz, puede producir que el algoritmo de conservación de volumen sobreestime los valores integrales necesarios para el balance de volumen.

El algoritmo desarrollado para la conservación de volumen funciona correctamente garantizando la conservación del volumen en pocas iteraciones, resultando computacionalmente muy económico. En general se ha comprobado como 3 o 4 iteraciones son suficientes para obtener un error relativo inferior a  $1e-5$ .

Un aspecto más a comentar es la sencillez de la generalización del método Level Set al caso tridimensional. Dado la formulación matemática del método, independiente de cualquier tipo de algoritmo geométrico, este proceso es trivial y no requiere de algoritmos especiales dimensionalmente dependientes.

Como se presenta en el ejemplo 8.5, los resultados obtenidos en el caso tridimensional son de alta calidad, demostrándose la robustez y consistencia de la formulación del método.

Se plantea una pregunta abierta que invita a futuras reflexiones:

¿Es el método Level Set una solución deseable al problema de advección?

Las dificultades presentes en la evolución de interfaces son debidas en gran medida al esquema de integración numérica de la ecuación de advección, (9.1). Estas dificultades son acentuadas cuando la función ha transportar presenta una discontinuidad de salto, como es el caso de la función escalón.

El método Level Set propone una solución “cosmética” al problema, convirtiendo una función escalón en una función distancia con mejores propiedades numéricas para su transporte. No obstante, no todos los problemas numéricos han sido solucionados

(redondeo en las equinas, pérdida de masa...). Con este proceso de regularización el esquema de integración numérica de (9.1) no se ha modificado en ningún estadio del método.

La motivación de la pregunta planteada se clara ahora: La solución del problema de advección pasa por métodos tipo Level Set o por mejorar el esquema de integración numérica de la ecuación de advección.

Durante el desarrollo de este estudio, se contemplo la posibilidad de modificar el esquema de integración numérica de (9.1). En particular se centraron esfuerzos en la modificación del término de estabilización de la ecuación de transporte. Como se vio en 6.1.1 este término se introduce en la ecuación en forma de difusividad adicional.

La modificación que se planteo fue reducir esta difusividad basándose en la curvatura de la interfaz, trivialmente calculable según (2.3), y el tamaño de elemento.

Los resultados obtenidos fueron buenos para ciertos ejemplos pero en general fueron no satisfactorios.

A pesar de los resultados obtenidos este tipo de estrategia no queda en absoluto descartada y se plantea como una de las futuras líneas de investigación.

Por ultimo, observar que en aquellas aplicaciones en que es necesario la solución de las ecuaciones de Navier-Stokes para obtener el campo de velocidades, queda gran trabajo que realizar. Dado que el campo de presiones obtenido no es correcto, se necesitan técnicas adicionales para su correcta interpolación. Este problema es también una de las futuras líneas de desarrollo.



## 10. Referencias

- [1] Adalsteinsson, D., Kimmel, R., Malladi, R., and Sethian, J.A., *Fast Marching Methods for Computing Solutions to Static Hamilton-Jacobi Equations*, report, Center for Pure and Applied Mathematics, PAM-667 (1996).
- [2] Adalsteinsson, D., and Sethian, J., *A Fast Level Set Method for Propagating Interfaces*, J. Comp. Phys., 118, pp. 269-277 (1995).
- [3] Codina R. *Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods*. Computer Methods in Applied Mech. and Eng. (2000); 190:1579–1599.
- [4] Codina R. *Stabilized Finite element approximation of transient incompressible flows using orthogonal subscales*. Computer Methods in Applied Mechanics and Engineering (2002); at press.
- [5] Codina R, Soto O. *A numerical model to track two-fluid interfaces based on a stabilized finite method and the level set technique*. Int. J. N. Methods in Fluids (2002): 40:293-301.
- [6] Codina R, Schafer U, Oñate E. *Mould filling simulation using finite elements*,. International Journal of Numerical Methods for Heat and Fluid Flow (1994); 4:291– 310.
- [7] Chang Y. C., Hou T. Y., Merriman, B. and Osher S., *A level set formulation of Eulerian interface capturing methods for incompressible fluid flows*, J. Comput. Phys. 124, 449 (1996).
- [8] Enright D, Fedkiw R, Freziger J, Mitchell I. *A Hybrid Particle Level Set Method for Improved Interface Capturing*. 2002.
- [9] Fedkiw R, Liu X.-D., *The ghost fluid method for viscous flows*, in *Progress in Numerical Solutions of Partial Differential Equations*, edited by M. Hafez (Arcachon, 1998).
- [10] Idelshon S, Oñate E, Ransau S. *A finite calculus formulation of the level set equation*. (2003).
- [11] Osher, S. and Sethian, J.A., *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, J. Comp. Phys., 79,1, pp. 12-49, (1988).
- [12] Osher S., *A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations*, SIAM J. Anal. 24, 1145 (1993).
- [13] Osher S, Fedkiw RP. *Level set methods: and overview and some recent results*. Journal of Computational Physics (2001); 169:463–502.
- [14] Oñate E, Miquel J, Hauke G. *A stabilized finite element method for the one dimensional advection-diffusion-absorption equation using finite calculus*. (2003)
- [15] Sethian, J.A., *A marching level set method for monotonically advancing fronts*, Proc. Nat. Acad. Sci., 93(4), (1996).
- [16] Sethian J. A., *Fast marching methods*, SIAM Rev. 41, 199 (1999).
- [17] Sethian J. A., *Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts*. Journal of Computational Physics 169, 503–555 (2001)
- [18] Sussman, M., Smereka, P., & Osher, S.J., *A level set approach for computing solutions to incompressible two-phase flows*, J. Comp. Phys., 94, 146-159 (1994).
- [19] Sussman M., Fatemi E, Smereka P, and Osher S, *An improved level set method for incompressible two-phase flows*, Comput. and Fluids 27, 663 (1998).
- [20] Sussman M, Smereka P, and Osher S, *A level set approach for computing solutions to incompressible two-phase flows*, J. Comput. Phys. 114, 146 (1994).
- [21] Taubin G. *A signal processing approach to fair surface design*. IBM Research Report, (1995).
- [22] Unverdi, S.O. and Tryggvason, G., *A Front-Tracking Method for Viscous, Incompressible, Multi-fluid Flows*, J. Comp. Phys., 100, pp. 25-37, (1992).
- [23] Zienkiewicz, O.C. and Taylor, R.L. *The Finite Element Method. Volume 3*. 5th Edition, Butterworth-Heinemann, 2001.

- [24] Sethian J. A., [http://math.berkeley.edu/~sethian/level\\_set.html](http://math.berkeley.edu/~sethian/level_set.html). *Fast marching and Level Set Methods*. Web page.